卐 वर दे वीणा वादिनी 卐

Contents

IMAGE PROCESSING AND COMPUTER VISION

UNIT 1

INTRODUCTION TO COMPUTER VISION AND IMAGE PROCESSING (CVIP) – BASICS OF CVIP, HISTORY OF CVIP, EVOLUTION OF CVIP, CV MODELS

Q.1. Explain the basics of computer vision and image processing (CVIP).

Ans. Computer vision can be defined as a scientific field that extracts information out of digital images. It has been expanded into wide area of field ranging from recording raw data into extraction of image pattern and information interpretation. It has a combination of concepts, techniques and ideas from digital image processing, pattern recognition, artificial intelligence and computer graphics. Most of the tasks in computer vision are related to the process of obtaining information on events or descriptions, from input scenes (digital images) and feature extraction. The methods used to solve problems in computer vision depend on the application domain and the nature of the data being analyzed.

Basically, computer vision is a combination of image processing and pattern recognition. The output of the computer vision process is image understanding. Development of this field is done by adapting the ability of human vision in taking information. Computer vision is the discipline of extracting information from images, as opposed to computer graphics. The development of computer vision is dependent on the computer technology system, whether about image quality improvement or image recognition. There is an overlap with image processing on basic techniques, and some authors use both terms interchangeably. The main purpose of computer vision is to create models and extracts data and information from images, while image processing is about implementing computational transformations for images, such as sharpening, contrast, among others. It also has similar meaning and sometimes overlapping with in human and computer interaction (HCI). Its coverage focus on more broad design, interface and all aspects of technologies related to interaction between human and computer. HCI is then developed as a separate discipline (which is the field of inerdisciplinary science) which discusses the

interrelationships between human-computer mediated by technology development including human aspects. Functionally, computer vision and human vision are the same, with the aim of interpreting spatial data i.e. data indexed by more than one dimension. However, computer vision cannot be expected to replicate just like the human eye.

Because of this the computer vision system has limited performance and function compared to human eye. Even though many scholars have proposed wide area of computer vision techniques to replicate human eye, however, in many cases, there are many limitations of the performance of computer vision system. One of the major challenges in their technique is the sensitivity of the parameters, the strength of the algorithm, and the accuracy of the results. It impacts on the complexity of performance evaluation of computer vision systems. Generally, the performance evaluation involves measuring some of the basic behaviours of an algorithm to achieve accuracy, strength, or extensibility to control and monitor system performance. The computer vision at the intersection of multiple scientific fields is shown in fig. 1.1.

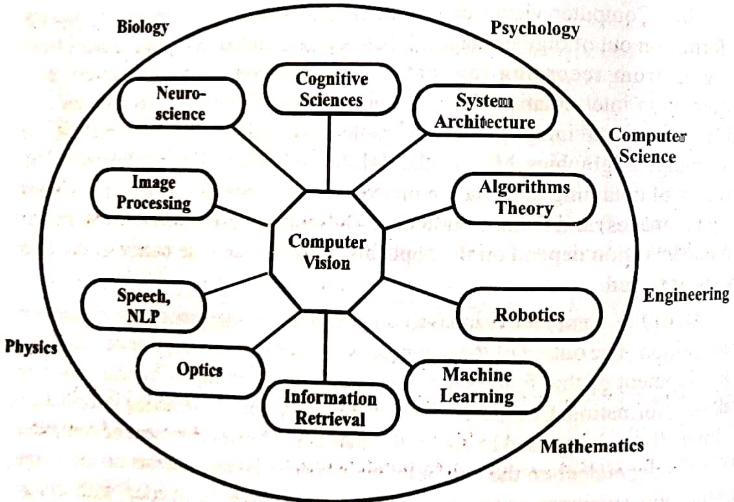


Fig. 1.1 Computer Vision

Q.2. What is digital image processing ?

Ans. Generally, digital image processing refers to processing of a 2-D picture by a digital computer.

In case of broader context, it implies digital processing of any 2-D data. DIP has a broad spectrum of applications, like radar, sonar, medical processing, image transmission and storage for business applications, acoustic image processing, remote sensing via satellites and other spacecrafts and automated inspection of industrial parts.

There are two principal of image processing –

(i) Improving image quality.

(ii) Machine perception of visual information.

For performing image processing, first digitize a picture into an image file. Then digital technique is applied to rearrange image puts, to improve the quality of shading or to improve colour separations. An example of enhance the quality of a picture is appeared in fig. 1.2. Similar techniques are employed to analysis galaxies images.

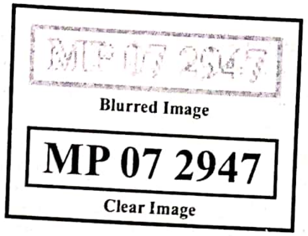


Fig. 1.2 Digital Image Processing

Q.3. Describe history of CVIP.

Ans. In between 1960-1970, when computer vision first started out in the early 1970's, it was viewed as the visual perception component of an ambitious agenda to mimic human intelligence and to endow robots with intelligent behaviour. At that time, it was believed by some of the early pioneers of artificial intelligence and robotics that solving the visual input problem would be an easy step along the path of solving more difficult problems such as higher level reasoning and planning. According to one well known story, in 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to spend the summer linking a camera to a computer and getting the computer to describe what it saw.

In 1980s to 1990s, a lot of attention was focused on more sophisticated mathematical techniques for performing quantitative image and scene analysis. Image pyramids started being widely used to perform tasks such as image blending and coarse to fine correspondence search.

In 1990s to 2000s, while a lot of the previously mentioned topics continued to be explored, a few of them became significantly more active. A burst of activity in using projective invariants for recognition evolved into a concentrated effort to solve the structure from motion problem.

From 2000s to till today, this past decade has continued to see a deepening interplay between the vision and graphics fields. In particular, many of the topics introduced under the rubric of image-based rendering such as image stitching, light-field capture and rendering etc.

The generation of computer vision is shown in fig. 1.3.

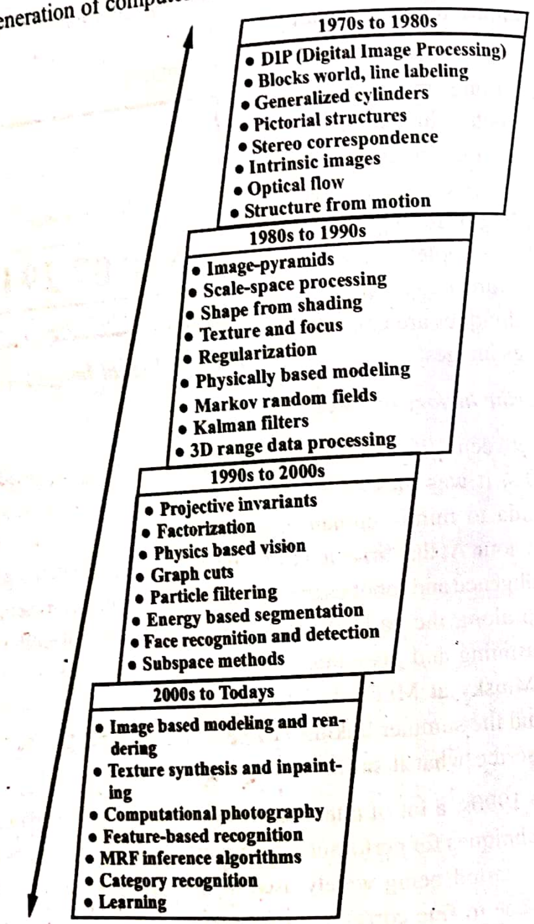


Fig. 1.3 Most Active Topics in CVIP

Q.4. Explain in detail about the operational step of CVIP.

Ans. Computer vision works by using algorithm and optical sensors to stimulate human visualization in order to automatically extract valuable information from an object. Compared to conventional methods that take a long time and require complex laboratory analysis, computer vision has been expanded into a branch of artificial intelligence (artificial intelligence) and simulated human visualization. It also combined with lighting systems to facilitate image acquisition continued with image analysis.

Image processing involves a series of image operations to enhance the quality of a digital image so as to remove defects such as geometric distortion, improper focus, repetitive noise, non-uniform lighting and camera motion. Image analysis is the process of distinguishing the objects from the background and producing quantitative information, which is subsequently used for decision making. Processing and analysis can be performed on many different types of image data. These include, in an increasing order of complexity such as, binary images, grayscale, color, polarized-light, multi-spectral and hyper-spectral, 3D images, multi-sensor and multimedia systems, and image sequences and video.

The image processing consist three levels of processing, viz. (i) low level processing which includes image acquisition and pre-processing of image, (ii) intermediate level processing which involves image segmentation, image representation and description, and (iii) high level processing which involves recognition of ROIs (regions of interests) and interpretation for quality sorting and grading. The terms machine vision or computer vision is often used for the entire subject, including image processing and analysis and pattern recognition techniques. Hence, the process of making a decision involves a number of steps in sequential order. Not all situations require all of these steps or operations, but all are potentially available to deal with particular problems.

Computer vision generally consists of the following five steps or operations as shown in fig. 1.4. First, image acquisition operations to convert images into

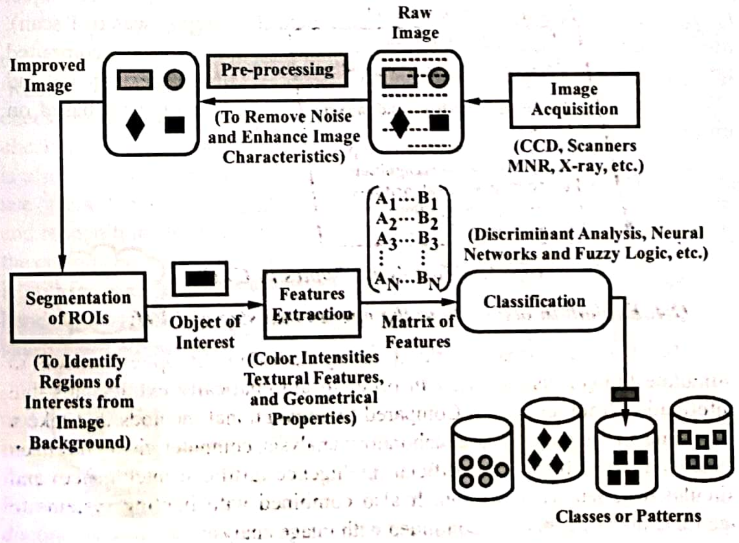


Fig. 1.4 The Operational Steps for CVIP

digital form. Second, pre-processing operations to obtain an improved image with the same dimensions as the original image. Third, image segmentation operations to partition a digital image into disjoint and non-overlapping regions. Fourth, object measurement operations to measure the characteristics of objects, such as size, shape, color and texture; and fifth, classification or sorting operations to identify objects by classifying them into different groups.

Q.5. Explain some examples of application using digital image processing.

Ans. Some examples of application using digital image processing are as follows –

(i) In X-ray Imaging – X-ray imaging is perhaps the most familiar type of imaging. X-ray is among the oldest sources of EM radiation used for imaging. The use of X-rays is medical diagnostics, but they also are used extensively in industry and other areas like astronomy. X-ray for medical imaging are generated using X-ray tube, which is a vacuum tube with a cathode and anode. The cathode is heated, causing free electrons to be released. These electrons flow at high speed to the positively charged anode. When the electrons strike a nucleus, energy is released in the form of X-ray radiation.

(ii) In Health Care – Several medical tools use image processing for various purposes, such as image enhancement, image compression, object recognition, etc. X-radiation (X-rays), computed tomography scan (CT scan), positron-emission tomography (PET), single-photon emission computed tomography (SPECT), nuclear magnetic resonance (NMR) spectroscopy and ultra-sonography are some popular pieces of medical equipment based on image processing.

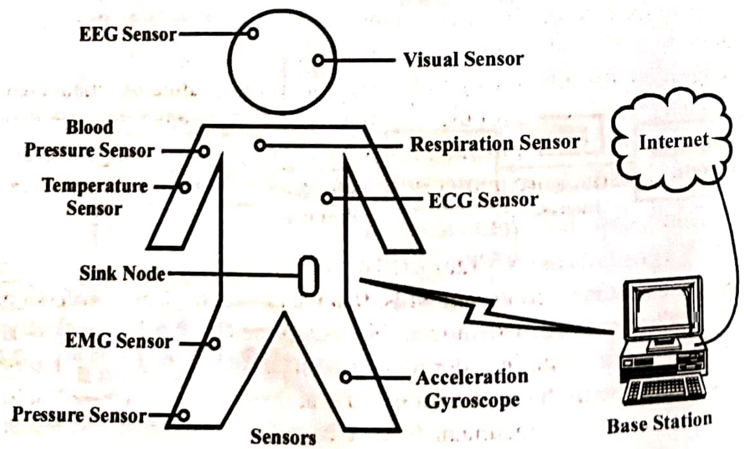


Fig. 1.5 Body Area Sensor Network

(iii) In Agriculture – Image processing, plays a vital role in the field of agriculture. Various paramount tasks such as weed detection, food grading, harvest control and fruit picking are done automatically with the help of image processing. Irrigated land mapping, determination of vegetation indices, canopy measurement, etc., are possible with good accuracy through the use of imaging techniques in various spectrums, such as hyper spectral imaging, infrared, etc.

(iv) In Weather Forecasting – Image processing also plays a crucial role in weather forecasting, such as prediction of rainfall, hailstorms, flooding. Meteorological radars are widely used to detect rainfall clouds and, based on this information, system predict immediate rainfall intensity.

(v) In Photography and Film – Retouched and spliced photos are extensively used in newspapers and magazines for the purpose of picture quality enhancement. In movies, many complex scenes are created with image and video editing tools which are based on image and video processing operations. Image processing-based methods are used to predict the success of upcoming movies. For a global media and entertainment company, latent view extracted over 6000 movie posters from IMDB along with their metadata (genre, cast, production, ratings, etc.), in order to predict the movies' success using image analytics. The colour schemes and objects in the movie posters were analyzed using machine learning (ML) algorithms and image processing techniques.

(vi) In Banking and Finance – The use of image processing-based techniques is rapidly increasing in the field of financial services and banking. 'Remote deposit capture' is a banking facility that allows customers to deposit checks electronically using mobile devices or scanners. The data from the check image is extracted and used in place of a physical check. Face detection is also being used in the bank customer authentication process. Some banks use 'facial-biometric' to protect sensitive information. Signature verification and recognition also plays a significant role in authenticating the signature of the customers. However, a robust system used to verify handwritten signatures is still in need of development. This process has many challenges because handwritten signatures are imprecise in nature, as their corners are not always sharp, lines are not perfectly straight, and curves are not necessarily smooth.

(vii) In Forensics – Tampered documents are widely used in criminal and civil cases, such as contested wills, financial paper work and professional business documentation. Documents like passports and driving licenses are frequently tampered with in order to be used illegally as identification proof. Forensic departments have to identify the authenticity of such suspicious documents. Identifying document forgery becomes increasingly challenging

due to the availability of advanced document-editing tools. The forger uses the latest technology to perfect his art. Computer scan documents are copied from one document to another to make them genuine. Forgery is not only confined to documents, it is also gaining popularity in images. Imagery has a remarkable role in various areas, such as forensic investigation, criminal investigation, surveillance systems, intelligence systems, sports, legal services, medical imaging, insurance claims and journalism.

(viii) In Security – Biometric verification systems provide a high level of authenticity and confidentiality. Biometric verification techniques are used for recognition of humans based on their behaviours or characteristics To create alerts for particularly undesirable behaviour, video surveillance systems are being employed in order to analyze peoples' movements and activities. Several banks and other departments are using these image processing-based video surveillance systems in order to detect undesired activities.

Q.6. Explain some applications of computer vision.

Ans. Applications of computer vision are as follows –

(i) Optical Character Recognition – This is, one of the oldest successful applications of computer vision to recognize characters and numbers. This can be used to read zipcodes, or license plates.

(ii) Mobile Visual Search – With computer vision, we can do a search on Google using an image as the query.

(iii) Self-driving Cars – Autonomous driving is one of the hottest applications of computer vision. Companies like Tesla, Google or General motors compete to be the first to build a fully autonomous car.

(iv) Automatic Checkout – Amazon Go is a new kind of store that has no checkout. With computer vision, algorithms detect exactly which products you take and they charge you as you walk out of the store.

(v) Vision-based Interaction – Microsoft's Kinect captures movement in real time and allows players to interact directly with a game through moves.

(vi) Augmented Reality – It is also a very hot field right now, and multiple companies are competing to provide the best mobile AR platform. Apple's ARKit has some impressive applications.

(vii) Scene Recognition – It is possible to recognize the location where a photo was taken. For instance, a photo of a landmark can be compared to billions of photos on google to find the best matches. We can then identify the best match and deduce the location of the photo.

(viii) Face Detection – It has been used for multiple years in cameras to take better pictures and focus on the faces. Smile detection can allow a camera to take pictures automatically when the subject is smiling. Face recognition is more difficult than face detection, but with the scale of today's data, companies like Facebook are able to get very good performance. Finally, we can also use computer vision for biometrics, using unique iris pattern recognition or fingerprints.

Q.7. Discuss about the image model in CVIP.

Ans. In image processing and computer vision, several image models have been accepted and are in recurrent use over the last several decades. In these image models, the emphasis is put on image quantities. The image support is formed by pixels usually considered as sampled locations. With each pixel is associated a scalar quantity called a gray level, or a vector quantity called either color at the perceptual level or multispectral at the signal level. Existing models differ primarily in terms of how the image quantity values are formulated and consequently the mathematical language used in the formulation. The well known image models are the function, the random process, and the ordered set. The image is a function $L_x \times L_y \to G^m$, where $L_x = \{1,\ldots,N_x\}$ and $L_y = \{1,\ldots,N_y\}$, $N_x \times N_y$ is the resolution of the image, $G = \{0,1,\ldots,n\}$, where n is the maximal quantity and m is the number of image bands. In the case of a binary image $n = 1$, image processing has roots in data structure, graph theory, language theory, logic, discrete geometry, and so on. If $n > 1$, usually the image is modeled as a real function (analog image where $G, L_x, L_y \subset \mathbb{R}$). In this case, function theory, functional analysis, differential equations, and differential geometry are the foundation. An image can also be represented as a collection of random variables $\{X(i, j) \mid (i, j) \in L_x \times L_y\}$. In this case, the probability density function, moments, sufficient statistics, time series, and the Markov processes are the roots. When the image is modeled as an ordered set, discrete mathematics and mathematical morphology are the foundation.

Fundamentally, an image is a physical or mathematical quantity where variables (image support) represent geometrical or temporal elements such as points, lines, surfaces, and time. Although all image models have deep roots in mathematics, the full functionality and role of the image support is not apparent and the association between the support and image quantities is not well defined. For a given computer vision or image processing task, no formal mechanism is given for the integration of physical topological, geometrical properties of objects and their precise functionality as part of the image model. Consequently, the resulting computational schemes are non-modular and sometimes not easy to reproduce. Our goal is to give a computational image model in terms of a data structure in which it is possible to retrieve all objects properties at any step of the processing to complete a

given task without overhead operations or drawback on the efficiency of the processing. This model can be seen as an abstract data structure in the sense that the image is the formal specification of the image variables, image quantities, the association between quantities and variables that allow to capture the geometrical and topological properties of objects as well as their physical and mathematical behaviour. This abstract view of the image as used in computer programs is defined by its attributes and a collection of meaningful operations. The attributes are the image support and quantities that are assigned to the image support such as the image radiometry (e.g., color and gray level) or any feature that can be deduced from the radiometry (e.g. texture). These quantities are scalar, vector or tensor. The allowable operations are of two kinds – the operations that are problem independent such as read and write and those that are problem dependent such as feature extraction, image segmentation, and image enhancement.

IMAGE FILTERING, IMAGE REPRESENTATIONS, IMAGE STATISTICS, RECOGNITION METHODOLOGY – CONDITIONING, LABELING, GROUPING, EXTRACTING AND MATCHING

Q.8. Write short note on image filtering.

Ans. Image filtering is done to improve the quality of the image. For example, smoothing an image reduces noise, blurred images can be rectified. There are broadly two types of algorithms – linear and non-linear. Linear filter is achieved through convolution and Fourier multiplication whereas non-linear filter cannot be achieved through any of these. Its output is not the linear function of its input thus, its result varies in a non-intuitive manner.

Filtering of image is an important process done in image processing. It can be done for noise removal, blur removal, edge detection etc. Linear and non-linear filters are the algorithms which are used for filtering. Right filter should be selected for any specific purpose. If the image or input given has less amount of noise but the magnitude is high then non-linear filters are used whereas linear low-pass filter is sufficient when the input given contains noise in large amount but the magnitude of noise is low. Linear filters are the most frequently used filters as it is simplest and fastest. Unlike non-linear filters, the linear filtering is done through applying the algorithm on the neighbour pixels of the input pixels in the image. The neighbourhood pixels are identified through their locations which are relative to the input pixel.

Q.9. Write short note on spatial filters.

Ans. Spatial filtering term is the filtering operations that are performed directly on the pixels of an image. The process consists simply of moving the filter mask from point to point in an image. At each point (x, y) the response of the filter at that point is calculated using a predefined relationship.

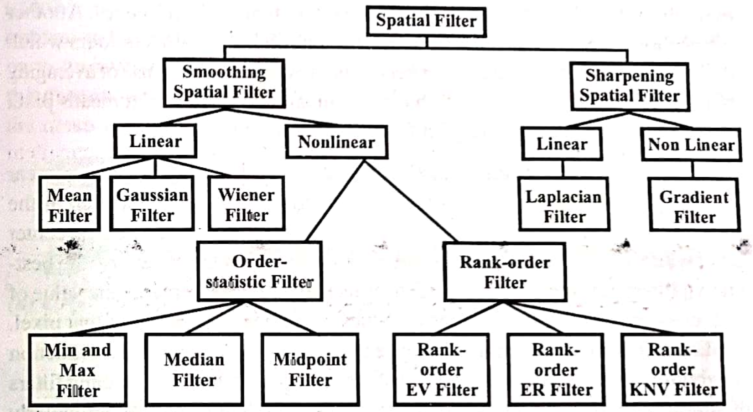


Fig. 1.6 Classification of Spatial Filter

Spatial filter can be classified into (i) smoothing spatial filters and (ii) sharpening spatial filters. These filters can be either linear or nonlinear. In linear filter each pixel value in the output image is a weighted sum of the pixel in the neighbourhood of the corresponding pixel in the input image. Nonlinear filtering operation is based conditionally on the values of the pixels in the neighbourhood, and they do not explicitly use coefficients in the sum-of-products manner. Noise reduction can be achieved effectively with a nonlinear filter.

Q.10. Explain in brief about smoothing spatial filters.

Ans. Smoothing filters are used for noise reduction and for blurring. Blurring is used in preprocessing tasks like removal of small details from an image prior to object extraction, and bridging of small gaps in lines or curves. By blurring with linear and nonlinear filters, noise reduction can be accomplished.

(i) Smoothing Linear Filters – The output of a smoothing, linear spatial filter is the average of the pixels contained in the neighbourhood of the filter mask.

The concept behind smoothing filters is straightforward. By replacing the value of every pixel in an image by the average of the intensity levels in the neighbourhood specified by the filter mask. The results of this process in an image with reduced "sharp" transitions in intensities. Due to random noise consists of sharp transitions in intensity levels, the most obvious smoothing application is noise reduction. Although, sharp intensity transitions characterize edges. So averaging filters have the undesirable side effect that they blur edges. Another application of this type of process is that the smoothing of false contours which result from using an insufficient number of intensity levels. A main use of averaging filters is in the reduction of irrelevant detail in an image. Irrelevant means pixel regions that are small with respect to the size of the filter mask.

(ii) Order-statistic (Nonlinear) Filters – These filters are nonlinear spatial filters whose response is based on ordering the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result. In this category, the best-known filter is the median filter, which, as its name implies, replaces the value of a pixel by the median of the intensity values in the neighbourhood of that pixel. Median filters are quite popular because they provide excellent noise-reduction capabilities with considerably less blurring as compare to linear smoothing filters of similar size for certain types of random noise. Median filters are particularly effective in the presence of impulse noise, also called salt and pepper noise because of its appearance as white and black dots superimposed on an image.

Q.11. Discuss the various types of smoothing linear filters.

Ans. The various types of smoothing linear filter are as follows –

(i) Mean Filter – The filter computes the value of each output pixel by finding the statistical mean of the neighbourhood of the corresponding input pixel. The following figure illustrates the local effect of the mean filter.

28	26	50
27	25	29
25	30	32

→ 30.22 → 30

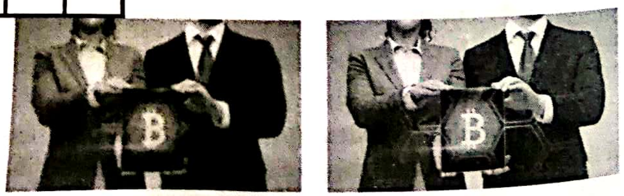
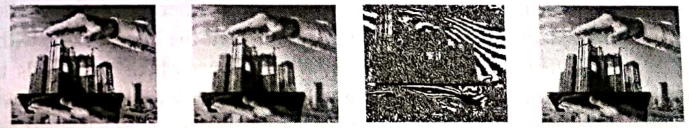


Fig. 1.7 The Effect of Mean Filter

The statistical mean of the neighbourhood on the left is passed as the output value associated with the pixel at the center of the neighbourhood.

(ii) Wiener Filter – The Wiener filter is a classic method for attempting to remove noise from images. It was developed (for 1D applications) by Norbert Wiener in the 1930's and 1940's. The Wiener filtering executes an optimal tradeoff between inverse filtering and noise smoothing. It removes the additive noise and inverts the blurring simultaneously. The Wiener filtering is optimal in terms of the mean square error. The Wiener filter has two separate parts, an inverse filtering part and a noise smoothing part. It not only performs the deconvolution by inverse filtering (high pass filtering) but also removes the noise with a compression operation (low pass filtering).



(a) Original Image (b) Image Blurred (c) Image after Inverse Filter (d) Image after the Wiener Filter

Fig. 1.8 Wiener Filter Applied to a Noise Image

Image statistics vary too much from a region to another even within the same image. Thus, both global statistics (mean variance, etc. of the whole image) and local statistics (mean, variance, etc. of a small region or sub-image) are important. Wiener filtering is based on both the global statistics and local statistics.

(iii) Gaussian Filter – A Gaussian filters smoothens an image by calculating weighted averages in a filter box. The Gaussian smoothing operator performs a weighted average of surrounding pixels based on the Gaussian distribution. It is used to remove Gaussian noise and is a realistic model of defocused lens.

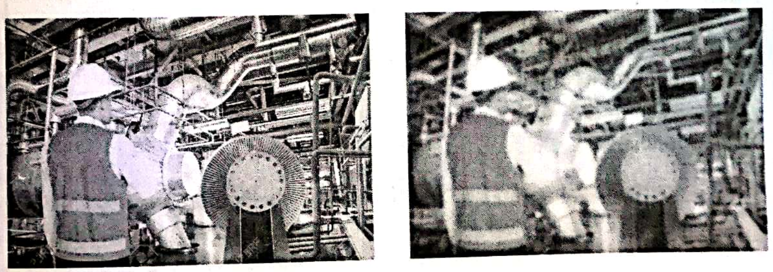


Fig. 1.9 Gaussian Smoothing

This removes fine image detail and noise leaving only larger scale changes. Gaussian Blurs produce a very pure smoothing effect without side effects. A Gaussian Blur is distinct from other blurs in that it has a well-defined effect on different levels of detail within an image.

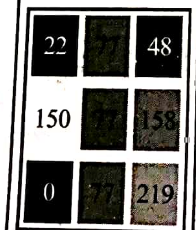
Q.12. Describe various types of nonlinear smoothing filters.

Ans. The various types of nonlinear smoothing filters are as follows –

(i) ***Order-statistic Filter*** – Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

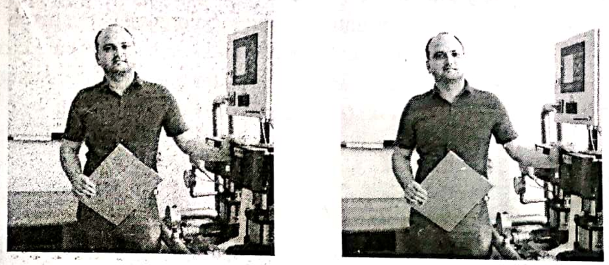
(a) Min and Max Filter – The minimum filter selects the smallest value within the pixel values and maximum filter selects the largest value within of pixel values. This is accomplished by a procedure which first finds the minimum and maximum intensity values of all the pixels within a windowed region around the pixel. If the intensity of the central pixel lies within the intensity range spread of its neighbours, it is passed on to the output image unchanged.

Table 1.1 The Example and Description of Max, Min and Midpoint Filters

Example Image	***Filter Type***	***Description***
	Max Filter	The center pixel would be changed from 77 to 219 as it is the brightest pixel within the current window.
	Min Filter	The center pixel would be changed from 77 to 0 as it is the darkest pixel within the current window.
	Midpoint Filter	The center pixel would be changed from 77 to 109 as it is the midpoint between the brightest pixel 219 and the darkest pixel 0 within the current window.

However, if the central pixel intensity is greater than the maximum value, it is set equal to the maximum value; if the central pixel intensity is less than the minimum value, it is set equal to the minimum value.

(b) Median Filter – Median filter is the nonlinear filter more used to remove the impulsive noise from an image. With the median filter, all the pixels in the neighbourhood are ranked by intensity level and the center pixel is replaced by that pixel which is mid-way in ranking.



(a) Noise Image *(b) Most of the Noise is Removed by using Median Filter*

Fig. 1.10

(c) Midpoint Filter – The midpoint filter blurs the image by replacing each pixel with the average of the highest pixel and the lowest pixel (with respect to intensity) within the specified window size.

Midpoint = (Darkest + Lightest)/2

(ii) ***Rank-order Filters*** – Rank-order filters are spatial-domain nonlinear filters, which are based on the correction of the local histogram within the filtering window. Rank-order filters are adaptive to the signal local statistics. Image processing with rank-order filters is reduced to the creation of the filtering interval from the limited number of pixels belonging to the filtering window with the further correction of the central pixel within the window using some kind of averaging of the selected pixels. There are three rank-order filters – Rank-order EV filter, Rank-order ER filter and Rank-order KNV filter.

(a) Rank-order EV Filter – A filtering interval for this filter is composed of all brightness values belonging to the pixels within the filtering window whose absolute difference from the central pixel brightness value is less than or equal to EV (which is a main control parameter for this filter). The most important property of this filter is that it smoothens the brightness jumps which are less than or equal to EV, and preserves the brightness jumps that are greater than EV. The EV filter is highly effective for reduction of white noise.

(b) Rank-order ER Filter – A filtering interval for this filter is composed of all brightness values belonging to the pixels within the filtering window whose rank difference from the central pixel brightness value rank in the variational series is less than or equal to ER, which is main control parameter for this filter. This filter is effective for the reduction of complicated noise types with unknown statistics, and for the reduction of any complicated noise containing an impulsive component.

(c) Rank-order KNV Filter – A filtering interval for this filter is composed of the number of brightness values (belonging to the pixels within the filtering window) which is equal to KNV and whose values are closest to the central pixel brightness value (KNV is a main control parameter for this filter). The most important property of this filter is that it smoothens only the objects whose area is less than the number of square pixels that are equal to KNV, and preserve the objects whose area is greater than KNV.

Q.13. Discuss about median filtering. ***(R.G.P.V., June 2017)***

Ans. The best known filter in order statistical nonlinear filters are known as median filters. Turkey introduces the concept of a median filter in 1997 which utilizing the median of the neighbourhood, to smoothen the image. Pratt extended this concept to two dimensional images. To determine each pixel value in the processed image following tasks are performed by the median filters –

(i) In the original image, all pixels in the neighbourhood of the pixel which are identified by the mask are sorted in descending or ascending order.

(ii) For the processed image, sorted median value is calculated and selected as the pixel value.

Median filter replaces the pixel value by the intensity median values in the neighbourhood of that pixel. For certain types of random noise, median filters have very good noise-reduction capabilities with considerably, minimum blurring as compared to similar size linear smoothing filters. In the presence of impulse noise, median filters are specially effective they are also known as ***salt-and-pepper noise*** because of its appearance as black and white dots superimposed on an image. The median ξ of a set of values is such that half the values in the set are equal or less than to ξ and half are equal or greater than to ξ. For performing median filtering at a point in an image, first sort the pixel values in the neighbourhood than find their median, and finally, assign that value to the corresponding pixel in the filtered image. For instance, the median is the 5^{th} largest value in a 3×3 neighbourhood and it is the 13^{th} largest value in a 5×5 neighbourhood. All equal values are grouped, when several values in a neighbourhood are the same. For instance, assume that a 5×5 neighbourhood has values (10, 20, 20, 20, 15, 20, 20, 25, 100, 12, 13, 14, 16, 17, 18, 19, 21, 22, 23, 24, 26, 27, 28, 29, 30). These values are sorted as (10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 20, 20, 20, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 100), which results in a median of 20. Hence, the median filters are used to force points with distinct intensity levels to be their neighbours. An $m \times m$ median filter eliminates isolated clusters of pixels which are light or dark with their neighbours, and whose area is less than $m^2/2$ (one half the filter area), here the word eliminated implies that forced to the median intensity of the neighbours. Larger clusters will be affected considerably less.

Fig. 1.11

Q.14. Write some properties of median filter.

Ans. There are several properties of median filter as follows –

(i) When the number of noise pixels in the window is larger than or half the number of pixels in the window, median filter performance is not good.

(ii) Median filter is a non linear filter. Hence for two functions x(k) and y(k) the relation is given below –

$$\text{median}\{x(k) + y(k)\} \neq \text{median}\ \{x(k)\} + \text{median}\ \{y(k)\}$$

(iii) While preserving spatial resolutions, median filter can be used for eliminating isolated lines or pixels.

Q.15. Explain in brief about sharpening spatial filters.

Ans. The sharpening is used to highlight transitions in intensity. Uses of image sharpening vary and include applications ranging from electronic printing and medical imaging to industrial inspection and autonomous guidance in military systems. Image blurring could be accomplished in the spatial domain by pixel averaging in a neighbourhood. It is logical to conclude that sharpening can be accomplished by spatial differentiation due to averaging is analogous to integration. This, in fact, is the condition, and deals with various ways of defining and implementing operators for sharpening by digital differentiation. Basically, the strength of the response of a derivative operator is proportional to the degree of intensity discontinuity of the image at the point at which the operator is applied. Hence, image differentiation enhances edges and other discontinuities and deemphasizes areas with slowly varying intensities.

We consider in some detail sharpening filters which are based on first and second order derivatives, respectively. We focus attention on one dimensional derivatives. In particular, we are interested in these derivatives behaviour in areas of constant intensity, at the onset and end of discontinuities, and along intensity ramps. Such discontinuities can be employed to model noise points, lines, and edges in an image. The behaviour of derivatives during transitions into and out of these image features also is of interest.

In terms of differences, the derivatives of a digital function are specified. There are multiple methods to define these differences. Although, it is required that any definition uses for a first derivative (i) must be zero in areas of constant intensity (ii) at the onset of an intensity step or ramp, must be nonzero (iii) must be nonzero along ramps. Similarly, any definition of a second derivative (i) in constant area, must be zero (ii) at the onset and end of an intensity step or ramp, must be nonzero (iii) must be zero along ramps of constant slope. Since we are dealing with digital quantities whose values are finite, the maximum possible intensity change also is finite, and the shortest distance over which that change can occur is between adjacent pixels.

A fundamental definition of the first-order derivative of a one dimensional function f(x) is the difference given below –

$$\frac{\partial f}{\partial x} = f(x + 1) - f(x) \qquad \text{...(i)}$$

Here, a partial derivative is used in order to keep the notation same as if we consider an image function of two variables, f(x, y), at the time of dealing with partial derivatives along the two spatial axes. Use of partial derivative does not affect in any way the nature of what we are trying to accomplish. Clearly, $\partial f/\partial x = df/dx$ if there is only one variable in the function and the same is true for the second derivative.

The second-order derivative of f(x) can be defined as the difference, which is given below –

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1) + f(x - 1) - 2f(x) \qquad \text{...(ii)}$$

It is easily verified that these two definitions satisfy the cases stated above.

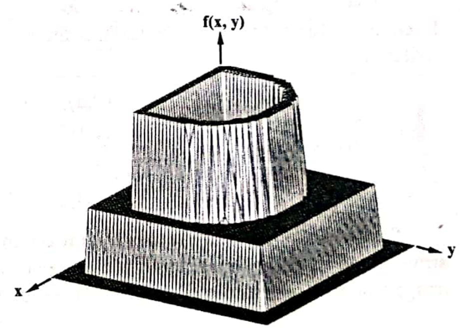
Q.16. Discuss about the image representation.

Ans. Image representation is the way in which images are described and stored in the computer. The efficiency of image processing algorithms will always be determined by the selection of different image representation methods to a great extent. Image representation is of primary importance for object recognition and image understanding. A good representation schema should be honest, general, brief and helpful for advanced tasks. As a fundamental data structure, a representation should capture the distribution of image features honestly and quickly, and make them accessible to higher processing layers.

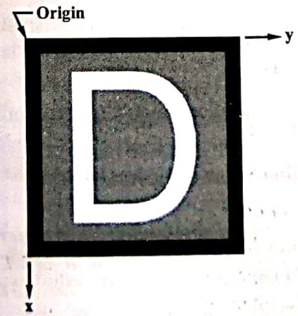
Q.17. Discuss the method of representing digital images.

Ans. Consider a continuous image function of two continuous variables, u and v is f(u, v). This function is converted into a digital image by using sampling and quantization. Assume that the image f(x, y) is a continuous in 2D array which has M rows and N columns, where (x, y) are discrete coordinates. We shall use integer value for these discrete coordinates, for national clarity and convenience. Thus, the value of the digital image at the origin is f(x, y) = f(0, 0). The next coordinate value along the first row of the image represented as f(x, y) = f(0, 1). Here, the notation (0, 1) is employed to signify the second sample along the first row. Normally, the value of the image is indicated f(x, y), at any coordinates (x, y) in which x and y are integers. The real plane part spanned by the coordinates of an images is known as spatial (time) domain, with m and n being referred to as spatial variables (spatial coordinates).

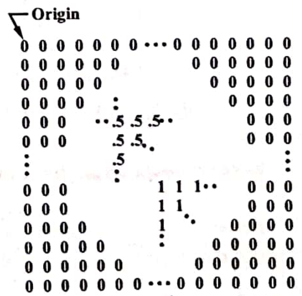
There are three basic ways to represent f(x, y) as shown in fig. 1.12. A plot of the function is shown in fig. 1.12 (a) with two axes finding spatial



(a)



(b)



(c)

Fig. 1.12

location and the third axis being the values of f as a function of the two spatial variables x and y. In fig. 1.12 (b), the representation is more common. It indicates f(x, y) as it would seam on a photograph or monitor. Here, each point intensity is proportional to the value of f at that point. There are only three equally spaced intensity values, as shown in fig. 1.12 (b). Each point in the image has the value 0, 0.5, or 1 when the intensity (gray level) is generalized to the interval [0, 1]. As shown in fig. 1.12 (b), a monitor or printer easily converts these three values to black, gray, or white respectively. The third representation is easily to show the numerical values of f(x, y) as an array. In this example, the size of f is 600 × 600 elements (360000 numbers). Obviously, printing the complete array should be bulky and communicate little information. However, if developing algorithms, this representation is quite useful when only sections of the image are printed and analyzed as numerical values. Fig. 1.12 (c) represents this concept graphically.

For processing and algorithm development, numerical arrays are used. In equation form, the representation of an M × N numerical array can be written as

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix} \quad ...(i)$$

The right side is a array or matrix of real numbers. Every element of this array is known as image element, picture element, pixel or pel. The words image and pixel are employed to represent a digital image and its elements –

$$\vec{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad ...(ii)$$

Obviously, $a_{ij} = f(x = i, y = j) = f(i, j)$, so equations (i) and (ii) are similar matrices.

Q.18. Explain elements of digital image processing system.

Or

Draw a neat block diagram representing components of a general purpose image processing system and explain each component in detail.

(R.G.P.V., Nov. 2019)

Ans. The basic components consisting a typical general purpose system used for digital image processing is shown in fig. 1.13.

There are following components of an image processing system as given below –

(i) Image sensors
(ii) Specialized image processing hardware
(iii) Computer
(iv) Image processing software
(v) Mass storage
(vi) Image displays
(vii) Hardcopy
(viii) Networking

(i) Image Sensors – Image sensing are required two elements to acquire digital image. The first is a physical device which is sensitive to the energy radiated by the object to image. The second is known as digitizer device which is converted the output of the physical sensing device into digital form. The sensors create an electrical output proportional to light intensity in a digital video camera. These outputs are converting into digital data with the help of digitizer.

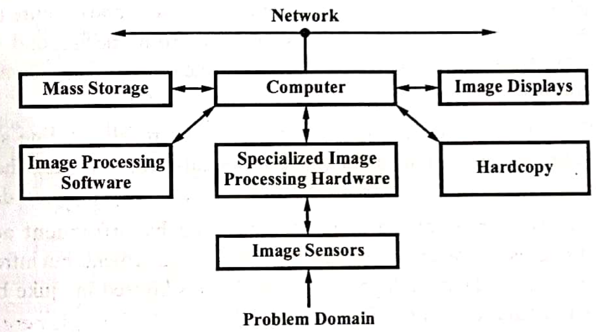


Fig. 1.13 Components of DIP System

(ii) Specialized Image Processing Hardware – It generally made up of the digitizer and hardware. The term hardware which is performed other primitive operations, like an arithmetic logic unit (ALU). ALU performs arithmetic and logical operations in parallel on whole images. Sometimes, this kind of hardware is known as front end subsystem and its most differentiating characteristic is speed.

(iii) Computer – The computers are used in digital image processing system. This term is also known as general-purpose computer and can range from personal computer to a super computer. Sometimes designed computers are employed to obtain a required performance level in dedicated applications, but our interest here is on general purpose digital image processing systems. In this systems, almost any well-equipped personal computer-type machine for off-line image processing operations is appropriate.

(iv) Image Processing Software – It made up of specialized modules which perform specific operations. For the user, a well-designed package also

includes the capability to write code that, as a minimum, utilizes the specialized modules. General-purpose software commands from at least one computer language and more sophisticated software packages permit the integration of those modules.

(v) Mass Storage Capability – It is a must in DIP applications. A size of image 1024 × 1024 pixels, where the gray (intensity) level of each pixel is an 8-bit quantity, needs one megabyte (1 MB = 10^6 byte) of storage space if the image is uncompressed. Providing sufficient storage in an image processing system may be a challenge when dealing with thousands of images. For image processing applications, a digital storage falls into three basic principal.

(a) During processing, short term storage is used. Computer memory is a technique of giving short term storage. Another is through specialized boards as known as frame buffers, which store one or more images and may be accessed rapidly, usually at video rates 30 frames/second. Frame buffers normally are housed in the specialized image processing hardware unit illustrated in fig. 1.13.

(b) On line storage for relatively fast recall, on line storage normally takes the form of magnetic disks or optical-media storage. The main factor characterizing on-line storage is frequent access to the stored data.

(c) Archival storage, characterized by infrequent access. Archival storage is characterized by massive storage requirements but infrequent require for access. Magnetic tapes and optical disks housed in "juke boxes" are the usual media for archival applications.

(vi) Image Displays – Image output is the final stage of the image processing system. Colour television (TV) monitors are used to display the image. Monitors are driven by the image outputs and graphics display cards which are an integral element of the computer system. For image display applications, seldom are there needed which cannot be met by display cards available commercially as element of the computer system. For some cases, these are implemented in the form headgear having two small displays embedded in goggles worn by the user and it is required to have stereo displays.

(vii) Hardcopy – It is used for recording images such as laser printers, heat-sensitive devices, film cameras, inkjet units and digital units, include optical and CD-ROM (compact disk-read only memory) disks. The highest possible resolution is provided by the film, but paper is the clear medium of selection for written material. When equipment of image projection is employed, images are shown on film transparencies or in a digital medium for presentations. For presentation of image, the latter procedure is gaining acceptance as the standard.

(viii) Networking – This is almost a default function in any computer system in use today. The consideration is bandwidth in image transmission due to the larger amount of data inherent in image processing applications. For dedicated networks, this typically is not a problem, but communications with remote sites through the Internet are not always as efficient. Luckily, this problem is enhancing quickly as a result of optical fiber and other broadband techniques.

Q.19. What are the fundamental steps in image processing ?

Ans. There are several steps in image processing as follows –

(i) Image Acquisition – Acquisition could be as easy as being provided an image which is already in digital form. In general, the image acquisition stage includes preprocessing like scaling.

(ii) Image Enhancement – It is the process of manipulating an image so that the result is more appropriate as compared to the original for a particular application. The word particular is important here, because it establishes at the outset that enhancement methods are problem oriented. Hence, for example, a procedure which is quite useful for enhancing X-ray images cannot be the best method for enhancing satellite images taken in the infrared band of the electromagnetic spectrum.

(iii) Image Restoration – It is an area which also deals with improving the appearance of an image. Although, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration methods tend to be based on mathematical or probabilistic models of image degradation. Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a "good" enhancement result.

(iv) Color Image Processing – It is an area that has been gaining in importance because of the significant increase in the use of digital images over the Internet.

(v) Wavelets – These are foundation for showing images in various degrees of resolution. Particularly, this is used for image data compression and for pyramidal representation, in which images are subdivided successively into smaller regions.

(vi) Compression – It deals with methods for reducing the storage needed to save an image, or the bandwidth needed to transmit it. However, storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. Particularly, this is true in uses of the Internet, that are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions like jpg file extension used in the JPEG image compression standard.

(vii) Morphological Processing – It deals with tools for extracting image components which are useful in the representation and description of shape.

(viii) Segmentation – It partition an image into its constituent parts or objects. Usually, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation method brings the process a long way toward successful solution of imaging problems which require objects to be identified individually.

(ix) Representation and Description – It always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. In either case converting the data to a form suitable for computer processing is necessary. The first decision which must be made is whether the data should be shown as a boundary or as a complete region. Boundary representation is appropriate if the focus is on external shape characteristics, like corners and inflections. Regional representation is appropriate if the focus is on internal properties, like texture or skeletal shape. These representations complement each other in some applications. Selecting a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. For describing the data, a procedure must also be specified so that features of interest are high lighted. Description, also known as feature selection. It deals with extracting attributes which result in some quantitative information of interest or are basic for differentiating one class of objects from another.

(x) Recognition – It is the process which assigns a label to an object based on its descriptors. We conclude our coverage of digital image processing with the development of procedures for recognition of individual objects. So far we have said nothing about the require for prior knowledge or about the interaction between the knowledge base and the processing modules in fig. 1.14. In the form of a knowledge database, knowledge about a problem domain is coded into an image processing system. This knowledge can be as easy as detailing regions of an image where the information of interest is known to be located, hence limiting the search that has to be conducted in seeking that information. The knowledge base can be quite complex, like an interrelated list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change detection applications. The knowledge base controls the interaction between modules to guiding the operation of each processing modules. The difference is made in fig. 1.14 by the use of double headed arrows between the processing modules and the knowledge base, as opposed to single headed arrows linking the processing modules.

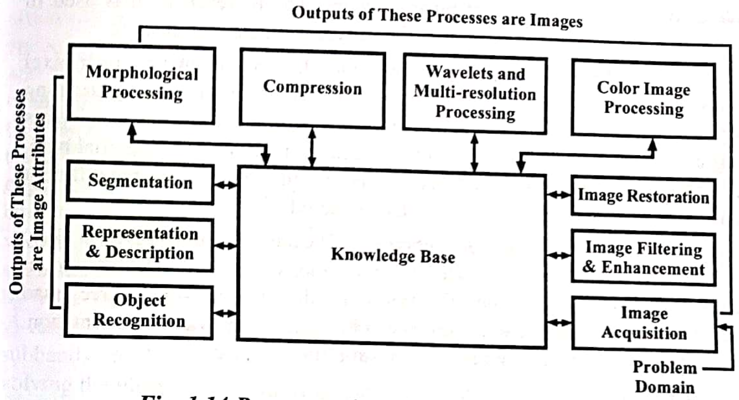


Fig. 1.14 Representation of Fundamental Steps

Q.20. Classify image representation methods based on level of processing.

Ans. Based on the level of processing of images by a machine for different purposes, the image representation methods are grouped into four categories, viz. pixel based, block based, region based and hierarchical based.

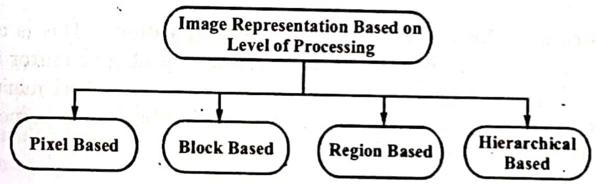


Fig. 1.15 Classification Based on Level of Processing

(i) Pixel Based Representation – This representation is the simplest representation to define an image. In digital imaging, a pixel, pel, or picture element is a physical point in a raster image, or the smallest addressable element in an all points addressable display device. The representation includes simple neighbourhood relations between elements. Each pixel contains only local information for each element. The number of elements in the representation is normally big and is used for displaying the image and it has applications in medical imaging where each pixel has got its own importance.

(ii) Block-based Representation – In this method, the image is divided in a set of (rectangular) array size. The number of elements is slightly smaller than with pixel-based, still only local information is stored which is same that of pixel based representations. Block based representations can be

(vii) Morphological Processing – It deals with tools for extracting image components which are useful in the representation and description of shape.

(viii) Segmentation – It partition an image into its constituent parts or objects. Usually, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation method brings the process a long way toward successful solution of imaging problems which require objects to be identified individually.

(ix) Representation and Description – It always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision which must be made is whether the data should be shown as a boundary or as a complete region. Boundary representation is appropriate if the focus is on external shape characteristics, like corners and inflections. Regional representation is appropriate if the focus is on internal properties, like texture or skeletal shape. These representations complement each other in some applications. Selecting a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. For describing the data, a procedure must also be specified so that features of interest are high lighted. Description, also known as feature selection. It deals with extracting attributes which result in some quantitative information of interest or are basic for differentiating one class of objects from another.

(x) Recognition – It is the process which assigns a label to an object based on its descriptors. We conclude our coverage of digital image processing with the development of procedures for recognition of individual objects. So far we have said nothing about the require for prior knowledge or about the interaction between the knowledge base and the processing modules in fig. 1.14. In the form of a knowledge database, knowledge about a problem domain is coded into an image processing system. This knowledge can be as easy as detailing regions of an image where the information of interest is known to be located, hence limiting the search that has to be conducted in seeking that information. The knowledge base can be quite complex, like an interrelated list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change detection applications. The knowledge base controls the interaction between modules to guiding the operation of each processing modules. The difference is made in fig. 1.14 by the use of double headed arrows between the processing modules and the knowledge base, as opposed to single headed arrows linking the processing modules.

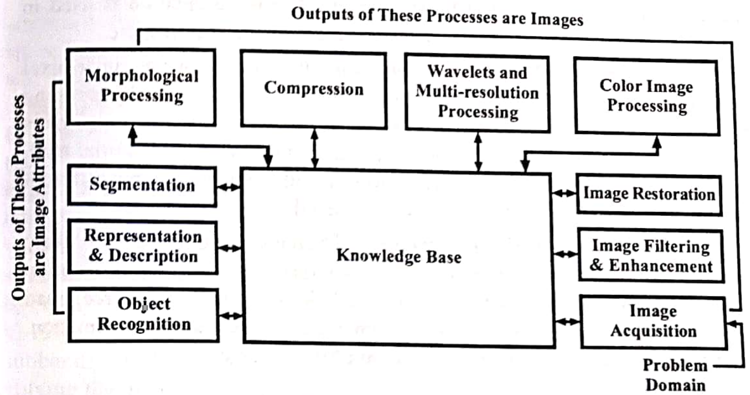


Fig. 1.14 Representation of Fundamental Steps

Q.20. Classify image representation methods based on level of processing.

Ans. Based on the level of processing of images by a machine for different purposes, the image representation methods are grouped into four categories, viz. pixel based, block based, region based and hierarchical based.

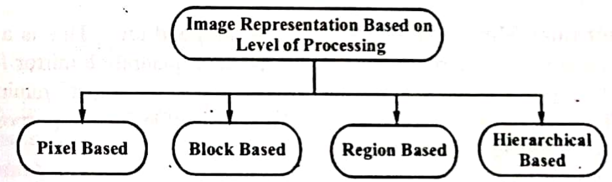


Fig. 1.15 Classification Based on Level of Processing

(i) Pixel Based Representation – This representation is the simplest representation to define an image. In digital imaging, a pixel, pel, or picture element is a physical point in a raster image, or the smallest addressable element in an all points addressable display device. The representation includes simple neighbourhood relations between elements. Each pixel contains only local information for each element. The number of elements in the representation is normally big and is used for displaying the image and it has applications in medical imaging where each pixel has got its own importance.

(ii) Block-based Representation – In this method, the image is divided in a set of (rectangular) array size. The number of elements is slightly smaller than with pixel-based, still only local information is stored which is same that of pixel based representations. Block based representations can be

done for both gray-scale and binary images. The representation is used in compression, segmentation, extracting different image features, etc.

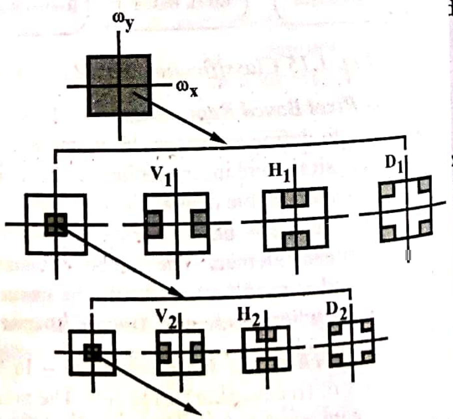
(iii) Region Based Representation – It is also known as super-pixel representation. Here the regions are not rectangular and it is formed by grouping similar and connected pixels. The adjacency information between regions is represented usually as RAG (region-adjacency graph) or combinatorial map. The representation is used for object detection and segmentation, but different unions of multiple regions have to be considered.

(iv) Hierarchical Representation – The representation uses most likely unions of regions of region-based representations. The image representation can be done at different scales. Examples includes min-/max-tree, α-tree, quad tree, bin tree, etc. Applications includes object detection, video segmentation, image segmentation and filtering, image simplification, etc.

Q.21. Explain in detail about the image statistics.

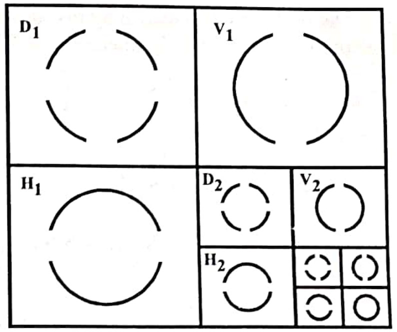
Ans. Two multiscale image decompositions, namely, the quadrature mirror filter (QMF) pyramid decomposition and the local angular harmonic decomposition (LAHD). These image statistics are collected from image representations that decompose an image using basis functions that are localized in both spatial positions and scales, implemented as a multi-scale image decomposition.

Quadrature Mirror Filter Pyramid Decomposition – This is a first multiscale image decomposition, based on separable quadrature mirror filters (QMF). One important reason for choosing this decomposition, is that it minimizes aliasing from the reconstructed image, making it suitable for the purpose of image analysis. In fig. 1.16 (a), an idealized frequency domain decomposition with a three scale QMF pyramid decomposition. From top to bottom, are scales 0, 1 and 2, and from left to right, are the low pass, vertical horizontal and diagonal subbands. And in fig. 1.16 (b), the magnitude of a three-scale QMF pyramid decomposition of a "disc" image. For the purpose of display, each subband is normalized into range 0 to 255.



(a) Three Scale QMF Pyramid Decomposition

The QMF pyramid decomposition splits the image frequency space into three different scales and within each scale, into three orientation subbands are vertical (V), horizontal (H) and diagonal (D). Visually each subband captures the local orientation energy in an image. The resulting vertical, horizontal and diagonal subbands at scale i are denoted by $V_i(x, y)$, $H_i(x, y)$ and $D_i(x, y)$ respec- tively. The first scale subbands are the result of convolving the image with a pair of 1-D $2n + 1$ tap finite impulse response (FIR) low-pass and high-pass QMF filters denoted as $l(.)$ and $h(.)$ respectively. The vertical subband is generated by convolving the image, $I(x, y)$, with the low-pass filter in the vertical direction and the high-pass filter in the horizontal direction as –



(b) The Magnitude of a Three-scale QMF Pyramid Decomposition of Disc Image

Fig. 1.16

$$V_1(x, y) = \sum_{m=-N}^{N} h(m) \sum_{n=-N}^{N} l(n) I(x-m, y-n)$$

The horizontal subband is generated by convolving the image with the low-pass filter in the horizontal direction and the high-pass filter in the vertical direction as –

$$H_1(x, y) = \sum_{m=-N}^{N} l(m) \sum_{n=-N}^{N} h(n) I(x-m, y-n)$$

The diagonal subband is obtained by convolving the image with the high-pass filter in both directions as –

$$D_1(x, y) = \sum_{m=-N}^{N} h(m) \sum_{n=-N}^{N} h(n) I(x-m, y-n)$$

Finally, convolving the image with the low-pass filter in both directions generates the residue low-pass subband, as –

$$L_1(x, y) = \sum_{m=-N}^{N} l(m) \sum_{n=-N}^{N} l(n) I(x-m, y-n)$$

The next scale is obtained by first down-sampling the residual low-pass subband L_1 and recursively filtering with $l(.)$ and $h(.)$, as

$$V_2(x, y) = \sum_{m=-N}^{N} h(m) \sum_{n=-N}^{N} l(n) L_1(\lfloor x/2 \rfloor - m, \lfloor y/2 \rfloor - n)$$

$$H_2(x, y) = \sum_{m=-N}^{N} l(m) \sum_{n=-N}^{N} h(n) L_1(\lfloor x/2 \rfloor - m, \lfloor y/2 \rfloor - n)$$

$$D_2(x, y) = \sum_{m=-N}^{N} h(m) \sum_{n=-N}^{N} h(n) L_1(\lfloor x/2 \rfloor - m, \lfloor y/2 \rfloor - n)$$

$$L_2(x, y) = \sum_{m=-N}^{N} l(m) \sum_{n=-N}^{N} l(n) L_1(\lfloor x/2 \rfloor - m, \lfloor y/2 \rfloor - n)$$

Subsequent scales are generated similarly by recursively decomposing the residual low-pass subband. The decomposition of a RGB color image is performed by decomposing each color channel independently. These sub-bands are denoted as $V_i^c(x,y), H_i^c(x,y)$, and $D_i^c(x,y)$, with $c \in \{r, g, b\}$. Color images using other color systems (e.g., HSV or CMYK) are decomposed by first transforming to RGB colors.

Local Angular Harmonic Decomposition – The another image decomposition is the local angular harmonic decomposition (LAHD). Formally, the n^{th}-order local angular harmonic decomposition of an image, I(x, y), is defined as –

$$A_n(I)(x,y) = \int_r \int_\theta I_{(x,y)}(r,\theta) R(r) e^{in\theta} dr\, d\theta$$

where $I_{(x,y)}(r,\theta)$ is the polar parameterization of image I(x, y) about point (x, y) in the image plane, and R(r) is an integrable radial function. The LAHD can be regarded as a local decomposition of image structure by projecting onto a set of angular Fourier basis kernels, $e^{in\theta}$. The function R(r) serves as the local windowing function as in the Gabor filters, which localizes the analysis in both the spatial and frequency domains. The output of the n-th LAHD, $A_n(I)(x, y)$, is a complex-valued 2-D signal. The magnitudes and phases of the first 4-order LAHD of an image is shown in fig. 1.18. Both the magnitudes and the phases capture image structures such as edges, corners and boundaries. Note that the basis in LAHD is highly over-complete and it is usually not possible to reconstruct the image from the decomposition.

Fig. 1.17 Original Image

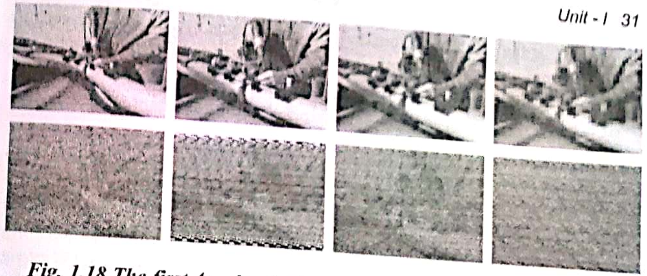


Fig. 1.18 The first 4-order LAHD of a Natural Image. The Top Row Shows the Magnitudes and the Bottom Row Shows the Phase Angles

Q.22. Describe the fundamental step in image recognition.

Ans. Image recognition is usually performed on digital images which are represented by a pixel matrix. The only information available to an image recognition system is the light intensities of each pixel and the location of a pixel in relation to its neighbours. From this information, image recognition systems must recover information which enables objects to be located and recognized, and, in the case of stereoscopic images, depth information which

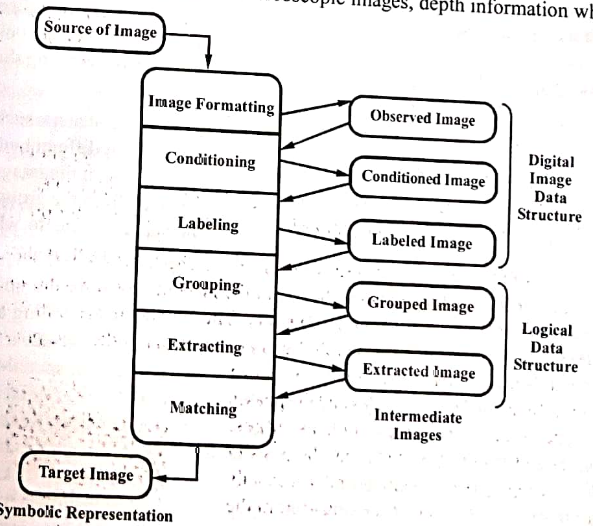


Fig. 1.19 Image Recognition

informs us of the spatial relationship between objects in a scene. The various steps required to transform iconic information into recognition information as shown in fig. 1.19.

(i) Image Formatting – This formatting means capturing an image by bringing it into a digital form.

(ii) Conditioning – In an image, there are usually features which are uninteresting, either because they were introduced into the image during the digitization process as noise, or because they form part of a background. An observed image is composed of informative patterns modified by uninteresting random variations. Conditioning suppresses, or normalizes, the uninteresting variations in the image, effectively highlighting the interesting parts of the image.

(iii) Labeling – In this step, informative patterns in an image have structure. Patterns are usually composed of adjacent pixels which share some property such that it can be inferred that they are part of the same structure (e.g., an edge). Edge detection techniques focus on identifying continuous adjacent pixels which differ greatly in intensity or colour, because these are likely to mark boundaries, between objects, or an object and the background, and hence form an edge. After the edge detection process is complete, many edge will have been identified. However, not all of the edges are significant. Thresholding filters out insignificant edges. The remaining edges are labeled. More complex labeling operations may involve identifying and labeling shape primitive and corner finding.

(iv) Grouping – In this step, labeling finds primitive objects, such as edges. Grouping can turn edges into lines by determining that different edges belong to the same spatial event. The first 3 operations represent the image as a digital image data structure (pixel information), however, from the grouping operation the data structure needs also to record the spatial events to which each pixel belongs. This information is stored in a logical data structure.

(v) Extracting – In this step, grouping only records the spatial event(s) to which pixels belong. Feature extraction involves generating a list of properties for each set of pixels in a spatial event. These may include a set's centroid, area, orientation, spatial moments, grey tone moments, spatial-grey tone moments, circumscribing circle, inscribing circle, etc.

Additionally properties depend on whether the group is considered a region or an arc. If it is a region, then the number of holes might be useful. In the case of an arc, the average curvature of the arc might be useful to know. Feature extraction can also describe the topographical relationships between different groups. Do they touch ? Does one occlude another ? Where are they in relation to each other ? etc.

(vi) Matching – In this step, once the pixels in the image have been grouped into objects and the relationship between the different objects has been determined, the final step is to recognize the objects in the image. Matching involves comparing each object in the image with previously stored models and determining the best match template matching.

MORPHOLOGICAL IMAGE PROCESSING – INTRODUCTION, DILATION, EROSION, OPENING, CLOSING, HIT-OR-MISS TRANSFORMATION, MORPHOLOGICAL ALGORITHM OPERATIONS ON BINARY IMAGES, MORPHOLOGICAL ALGORITHM OPERATIONS ON GRAY-SCALE IMAGES, THINNING, THICKENING, REGION GROWING, REGION SHRINKING

Q.23. Discuss briefly about mathematical morphology.

(R.G.P.V., June 2017)

Ans. Matheron and Serra had developed mathematical morphology technique at the Ecole des Mines in Paris. For developing this technique, the motivation comes from the structural information collection about the image domain. For extracting image elements, mathematical morphology is a tool. These image elements are useful to representation and description. Mathematical morphology content is fully depended on set theory. In mathematical morphology, there are several useful operators which specified by using set operations. Sets show objects in an image. The sets are the members of the 3-D image domain with their integer elements in a binary image. In a binary image, the black or white pixels like a set refer to the image morphological description. x and y coordinates are the elements of a 3-D tuple which represents each element in the image. Mathematical morphology plays a important role in procedures for image description and it can be used as the basis for developing image-segmentation methods with a wide range of applications.

Q.24. Explain about basic set theory.

Ans. Morphology is depend on set theory. Set theory includes various operations such as union, intersection, complement, difference, reflection, translation are given below –

(i) Union – $P \cup Q$ represents union of images P and Q. $P \cup Q$ represents the set whose elements can be elements of P and Q or either element of P or element of Q. The expression is written as

$$P \cup Q = \text{def}\{x | x \in P \text{ or } x \in Q\}$$

(ii) Intersection – $P \cap Q$ represents intersection of images P and Q. $P \cap Q$ represents the set whose elements are common for both image P and image Q. The expression is written as –

$$P \cap Q = \text{def}\{x \mid x \in P \text{ and } x \in Q\}$$

(iii) Complement – P^c represents complement of image P. P^c represents the set which involving everything not in image P. The expression is written as –

$$P^c = \text{def}\ \{x \mid x \notin P\}$$

(iv) Difference – P – Q represents difference between image P and image Q. P – Q represents the set which involves subtracts all elements of image P, which are in image Q. The expression is written as –

$$P - Q = \text{def}\{x \mid x \in P \text{ and } x \notin Q\}$$

(v) Reflection – $\hat{Q}$ represents reflection of image Q. The expression is –

$$\hat{Q} = \{w \mid w = -q \text{ for } q \in Q\}$$

(vi) Translation – $(P)_z$ represents translation of image P. The expression is –

$$(P)_z = \{c \mid c = p + z, \text{ for } p \in P\}$$

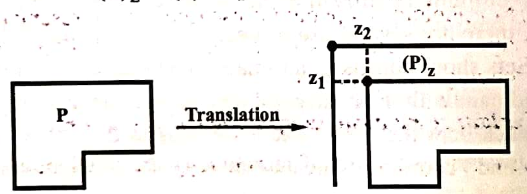


Fig. 1.20 Image P Translation

Q.25. Write a short note on binary morphology.

Ans. Binary morphology was used as a principal method because binary morphology gives general routines for pattern matching and it is fast, memory efficient. Today, processors are fast, memory is cheap and 10 year ago we can not be imagined but we are using binary morphology for pattern matching at fast speed. Binary morphology is related with sets operations. In a binary image, foreground pixels mean black pixels or ON or 1 and image background pixels mean white pixels or off or 0. In binary images, tuple or a 2-dimensional vector of the (x, y)-plane represents each element of binary image.

Q.26. What are the few applications of morphological based operations in image processing ?

Ans. The applications of morphological based operations are as follows –

(i) To remove noise in the image

(ii) To quantitative description of images

(iii) To segment images from the background

(iv) To enhance the image structure.

Q.27. Give the features of morphological operations.

Ans. There are several features of morphological operations as follows –

(i) When the size of the structural element expands, then morphological operations remove information of a greater extent.

(ii) Morphological operations uses a well-developed morphological algebra for representation and optimization.

(iii) This is possible to describe digital algorithm in the form of a very small class of primitive morphological operations.

(iv) While managing the stability of the important geometric characteristics, morphological operations give systematic alteration of the geometric content of an image.

(v) Non invertibility of morphological operations characterize their linear transformations.

(vi) Morphological operations use rigorous representation theorems, which one may get the expression of morphological filters in form of the primitive morphological operations.

Q.28. Explain some basic morphological operations.

Ans. The basic operations of morphology are dilation, erosion, closing and opening.

(i) Dilation – A dilation operation enlarges a region. A dilation adds pixels to the perimeter of each image object (sets their values to 1), filling in holes and broken areas, and connecting areas that are separated by spaces smaller than the size of the structuring element. The dilation (operator $\oplus$) is defined as –

$$\text{dilation}(x) = x \oplus s = \bigcup_{a \in s} x_a$$

(ii) Erosion – Erosion is an operator that basically removes objects smaller than the structuring element and removes perimeter pixels from the border of larger image objects (sets the pixel value to 0). If x is an image and s is the structuring element (mask), the erosion (operator $\ominus$) is defined as –

$$\text{erosion}(x) = x \ominus s = \bigcap_{a \in s} x_{-a}$$

where x_a indicates a basic shift operation in the direction of element 'a' of s and x_{-a} would indicate the reverse shift operation.

(iii) Opening – An opening operation (erosion then dilation) can separate objects that are connected in a binary image. Opening generally smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions. Mathematically, the opening function can be described by

$$\text{opening}(x) = \text{dilation}(\text{erosion}(x))$$

or, using the operator $\circ$,

$$x \circ s = (x \ominus s) \oplus s$$

(iv) Closing – The closing operation is defined as dilation followed by an erosion using the same structuring element. A closing operation can close up internal holes and gaps in a region and eliminate bays along the boundary.

$$\text{closing}(x) = \text{erosion}(\text{dilation}(x))$$

or, using the operator $\bullet$,

$$x \bullet s = (x \oplus s) \ominus s$$

Q.29. What is the use of dilation and erosion ?

Ans. **Use of Dilation** – It is used for bridging gaps. Simple structuring element is used to repairing the gaps. The maximum length of the gaps is called to be two pixels.

Use of Erosion – It is used for eliminating irrelevant detail from binary image. Structuring element is helped to eliminate irrelevant detail from a binary image.

Q.30. What is the properties of dilation and erosion ?

Ans. There are several properties of dilation and erosion as follows –

(i) Dilation and erosion are not inverses of each other.

(ii) The dilation and erosion are translation invariant.

(iii) Increasing –

If $\left.\begin{array}{ll} x \subset x' \Rightarrow x \ominus s \subset x' \ominus s & \forall s \\ x \oplus s \subset x' \oplus s & \forall s \end{array}\right\}$

If $s \subset s' \Rightarrow x \ominus s \subset x \ominus s' \quad \forall x$

(iv) Distributivity –

$$x \oplus (s \cup s') = (x \oplus s) \cup (x \oplus s')$$
$$x \ominus (s \cup s') = (x \ominus s) \cap (x \ominus s')$$

(v) Iteration –

$$\left.\begin{array}{l} (x \ominus s) \ominus s' = x \ominus (s \oplus s') \\ (x \oplus s) \oplus s' = x \oplus (s \oplus s') \end{array}\right\}$$

(vi) Local Knowledge –

$$(x \cap z) \ominus s = (x \ominus s) \cap (z \ominus s)$$

(vii) Duality –

$$x^c \oplus s = (x \ominus s)^c$$

According to this rule erosion and dilation are duals with respect to the complement operation.

(viii) Expansively –

$$x \geq x \ominus s \quad \text{(Erosion)}$$
$$x \leq x \oplus s \quad \text{(Dilation)}$$

The output image of dilation operation is expanded, so that dilation is expansive. But the output image of erosion operation is not expanded, so that erosion is anti-expansive.

(xi) Commutative –

$$x \oplus s = s \oplus x \quad \text{(Erosion)}$$
$$x \ominus s \neq s \ominus x \quad \text{(Dilation)}$$

Q.31. What are the properties of open and close operations ?

Ans. There are several properties of open operation as follows –

(i) $x \circ s$ is a subset of x.

(ii) When P is a subset of Q, then $P \circ s$ is a subset of $Q \circ s$.

(iii) Open is increase operation. Hence,

$$x \circ s \leq y \circ s$$

(iv) The open operation is anti-expansive because erosion is followed by dilation in open operation. Hence,

$$x \circ s \leq x$$

(v) The opening of x equivalents to the closing of the complemented image x^c.

$$x \circ s \leq (x^c \bullet s)^c$$

(vi) $$x \circ s = (x \circ s) \circ s$$

There are several properties of close operation as follows –

(i) x is a subset of $x \bullet s$.

(ii) If P is a subset of Q, then $P \bullet s$ is a subset of $Q \bullet s$.

(iii) Close is increase operation. Hence,

$$x \bullet s \leq y \bullet s$$

(iv) The close operation is expansive because dilation is followed by erosion in close operation. Hence,

$$x \leq x \bullet s$$

(v) The closing of x equivalents to the opening of the complemented image x^c.

$$x \bullet s \leq (x^c \circ s)^c$$

(vi) $$x \bullet s = (x \bullet s) \bullet s$$

Q.32. Explain how hit-or-miss transformation is used for finding local patterns of pixels. *(R.G.P.V., June 2015)*

Ans. A transformation which is employed for template matching is known as hit-or-miss transformation. It is a morphological operator which is used for searching structuring element size or local patterns of pixels. Hit-or-miss transformation investigates the inside and outside of images at the same time by using two different structuring elements. When the first structuring element translated to that pixel fits the image and the second structuring translation element misses object, then a pixel belonging to an object is preserved by the hit-or-miss operation. Two template sets s and (w-s) is included by the hit-or-miss transformation. These two sets are disjoint. Image background is matched by template (w-s) and image foreground is matched by template s. Intersection of the foreground erosion with s and the background erosion with (w-s) is the hit-or-miss transformation. Expression for the hit-or-miss transform can be written as –

$$HM(x, s) = (x \ominus s) \cap [x^c \ominus (w - s)]$$

Here, x = Input image
s = Structuring element
w = Small window which includes at least one pixel, thicker than s.

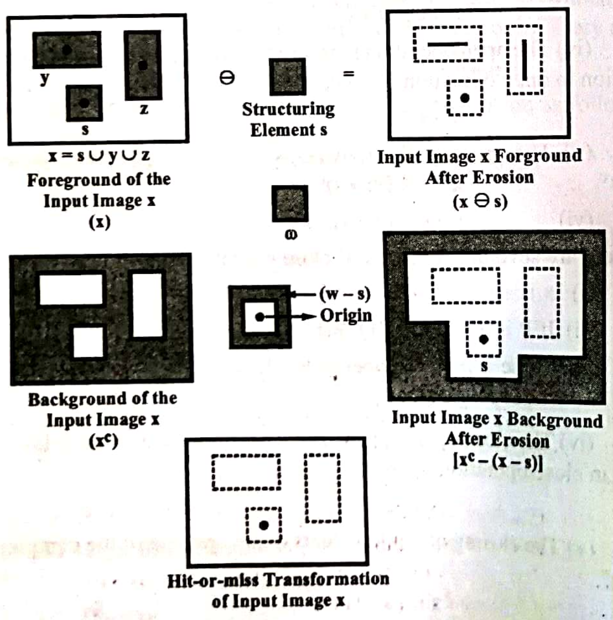
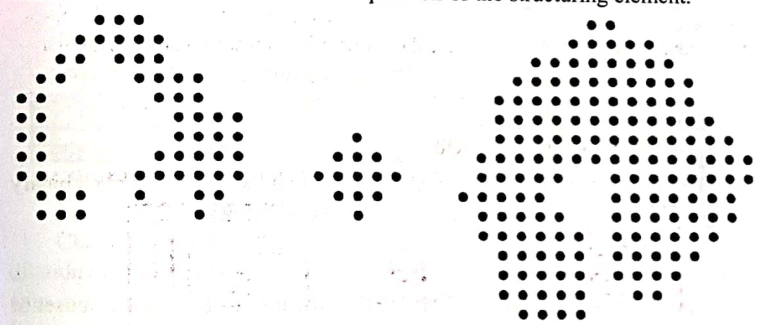


Fig. 1.21 Process of Hit-or-miss Transformation

Q.33. Explain morphological operation on binary image.

Ans. There are two basic morphological operations as follows –

(i) Dilation – The process of expand the binary image from its original shape is known as dilation. It is an expansion operation. It is also an expansion operator. It increases size of binary objects. Structuring element is used to determine the way of binary image expand. The size of structuring element is small as compare to original image, and usually the size which is used for the structuring element is 3 × 3. The structuring element is reflected and shifted from top to bottom and from left to right at each shift, the process will look. For any overlapping similar pixels between the structuring element and that of binary image. When there is any overlapping found between pixels then pixels will be turned to black or 1 under the centre position of the structuring element.



(a) Input Image (x) *(b) Structuring Element (s)* *(c) Dilated Image* $(x \oplus s)$

Fig. 1.22 Dilation

Let us assumes as the structuring element and x as the reference image. The expression of dilation operation is as follows –

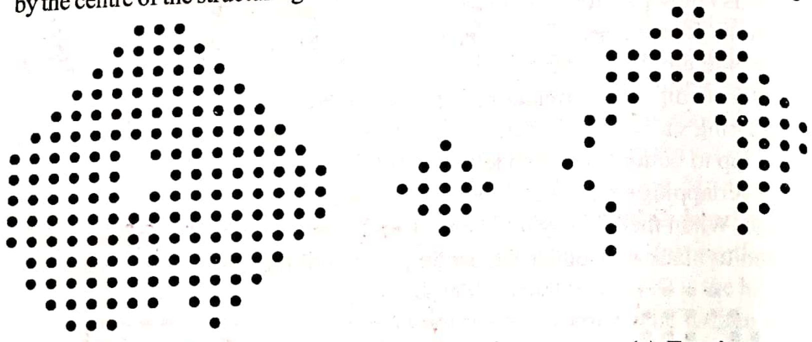
$$x \oplus s = \{z | [(\hat{s})_z \cap x] \subseteq x\}$$

Here, $\hat{s}$ = Images moved about the origin.

The above equation states that the outcome element z should be that there will be at least one element in s that intersects with an element in x when the structuring element dilates image x. When this is the condition, the place in which the structuring element is being centred on the image will be black or 1 or ON.

(ii) Erosion – Erosion is the process of decrease the binary image from its original shape. It is a thinning operation. It is also a thinning operator. It decreases size of binary objects. The structuring element is used to determine the way of binary image shrink. The size of structuring element is small as compare to original image, and usually the size which is used for the structuring element is 3 × 3. The erosion process can shift the element of structuring

from top to bottom and left-to-right. Presented by the structuring element centre at the centre position, the process will observe to whether there is a total overlap with element of structuring or not. The centre pixel represented by the centre of the structuring element is white (0) when there is no overlapping.



(a) Input Image (x) (b) Structuring Element (s) (c) Erosion ($x \ominus s$)

Fig. 1.23 Erosion

Let us assume s as the structuring element and x as the reference binary image. The expression of erosion operation is as follows –

$$x \ominus s = \{z|(s)_z \subseteq x\}$$

The above expression states that if the structuring element is a subset or equal to the binary image x then the output element z is taken only.

Q.34. Explain opening and closing operations on images by using suitable example. ***(R.G.P.V., June 2015)***

Or

With necessary figures, explain the opening and closing operations ? ***(R.G.P.V., Nov. 2018)***

Ans. **Opening Operation** – It is based on morphological operation such as erosion and dilation. This is erosion operation followed by a dilation. It is used to smooth the inside of the object contour, breaks narrow strips and removes thin portions of the image and to remove noise and CCD defects in the images. The mathematical expression of opening is written as –

$$x \circ s = (x \ominus s) \oplus s$$

In words, the opening of x by s is simply the erosion of x by s, followed by the dilation of the result by s.

where x represents input image and s represents structuring element.

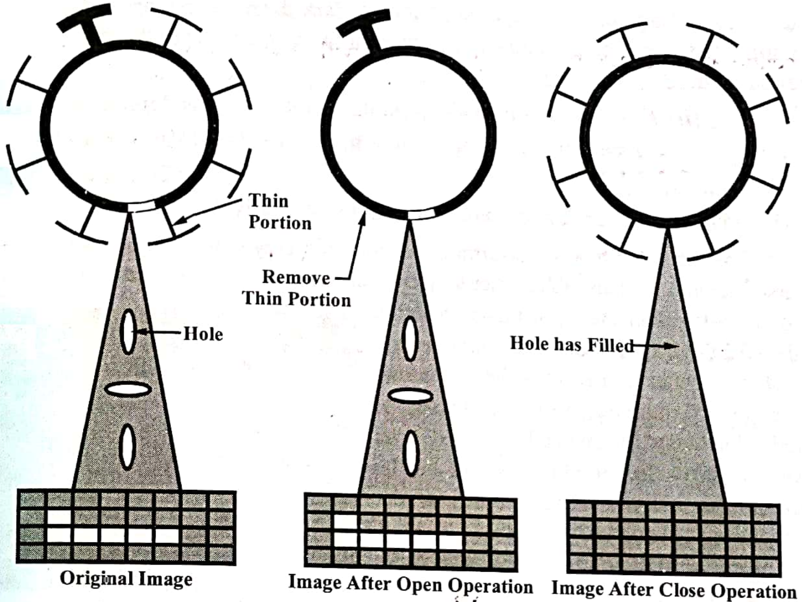


Fig. 1.24 Open and Close Operations

Closing Operation – It is opposite of the opening operation. This is dilation operation followed by an erosion. It is used to fill the small holes and gaps in a single pixels object. Closing also tends to smooth sections of contours. It manages the shapes and sizes of images. The mathematical expression of closing is written as –

$$x \bullet s = (x \oplus s) \ominus s$$

In words, the closing of x by s is simply the dilation of x by s, followed by the erosion of the result by s.

Where x represents input image and s represents structuring element.

Q.35. Explain morphological operations on gray-scale images.

Ans. Dilation, erosion, opening and closing are basic operations of morphology. We use these operations to develop several basic gray-scale morphological algorithms.

(i) Dilation – Gray-scale dilation of x by s, denoted $x \oplus s$ is defined as

$$x \oplus s(m,n) = \max\{x(m-p, n-q) + s(p,q) \mid (m-p), (n-q) \in D_x; (p,q) \in D_s\}$$

where, D_x and D_S are the domains of x and s, respectively.

The general effect of performing dilation on a gray scale image is twofold, first, if all the values of the structuring element are positive, the output image

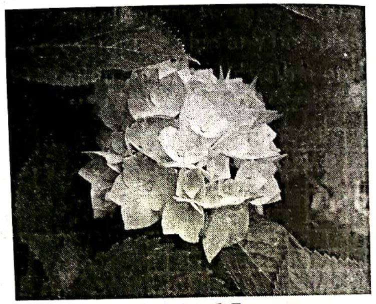
tends to be brighter than the input. Second, dark details either are reduced or eliminated, depending on how their values and shapes relate to the structuring element used for dilation.

(ii) Erosion – Gray-scale erosion, denoted $x \ominus s$ is defined as –

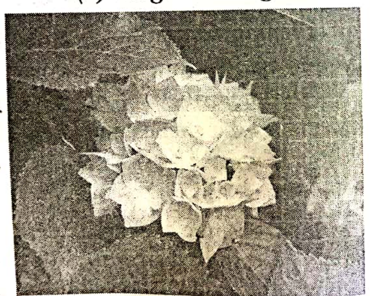
$$x \ominus s\ (m, n) = \min\{x(m+p, n+q) - s(p,q) \mid (m+p), (n+q) \in D_X; (p,q) \in D_S\}$$

where D_X and D_S are the domains of x and s respectively.

The general effect of performing erosion on a gray-scale image is twofold – first, if all the elements of the structuring element are positive, the output image tends to be darker than the input image. Second, the effect of bright details in the input image that are smaller in area than the structuring element is reduced, with the degree of reduction being determined by the gray-level values surrounding the bright detail and by the shape and amplitude values of the structuring element itself.



(a) Original Image



(b) Result of Dilation

(c) Result of Erosion

Fig. 1.25

Gray-scale dilation and erosion are duals with respect to function complementation and reflection. That is

$$(x \ominus s)^c\ (m, n) = (x^c \oplus \hat{s})(m,n)$$

where $x^c = -x\,(p, q)$ and $\hat{s} = s(-p, -q)$.

For example, take a simple-gray-scale image as shown in fig. 1.25 (a). In fig. 1.25 (b), the result of dilating this image with a 'flat top' structuring element in the shape of a parallel epiped of unit height and size 5 × 5 pixels. Dilation is expected to produce an image that is brighter than the original and in which small, dark details have been reduced or eliminated. In fig. 1.25 (c), the result of erosion is opposite effect to dilation. The eroded image is darker, and the size of small, bright features were reduced.

(iii) Opening – The opening of image x by subimage s, denoted $x \circ s$ is defined as –

$$x \circ s = (x \ominus s) \oplus s$$

As in the binary case, opening is simply the erosion of x by s. Followed by a dilation of the result by s.

The gray-scale opening operation satisfies the following properties –

(a) $(x \circ s) \lrcorner\, x$

(b) If $x_1 \lrcorner\, x_2$, then $(x_1 \circ s) \lrcorner\, (x_2 \circ s)$

(c) $(x \circ x) \circ x = x \circ s$

The notation $e \lrcorner\, r$ is used to indicate that the domain of e is a subset of the domain of r and also that $e(p,q) \le r(p,q)$ for any (p, q) in the domain of e.

(iv) Closing – The closing of x by s, denoted x.s is defined as –

$$x \bullet s = (x \oplus s) \ominus s$$

The closing operation satisfies the following properties –

(a) $x \lrcorner\, (x \bullet s)$

(b) If $x_1 \lrcorner\, x_2$, then $(x_1 . s) \lrcorner\, (x_2 . s)$.

(iii) $(x \bullet s) \bullet x = x \bullet x$.

The usefulness of these expressions is similar to that of their binary counterparts.

The opening and closing for gray-scale images are duals with respect to complementation and reflection. That is

$$(x \bullet s)^c = x^c \circ \hat{s}$$

where $x^c = -x(p, q)$, above equation can be written as – $(x \bullet s) = (-x \circ \hat{s})$.

Opening and closing of images have a simple geometric interpretation. Suppose that we view an image function x(p, q) in 3-D perspective, with the p- and q axes being the usual spatial coordinates and the third axis being gray-level values. In this representation, the image appears as a discrete surface whose value of any point (p, q) is the value of x at those coordinates. Suppose that we open x by a spherical structuring element, s viewing this element as a "rolling ball". Then the mechanics of opening x by s may be interpreted geometrically as the process of pushing the ball against the underside of the

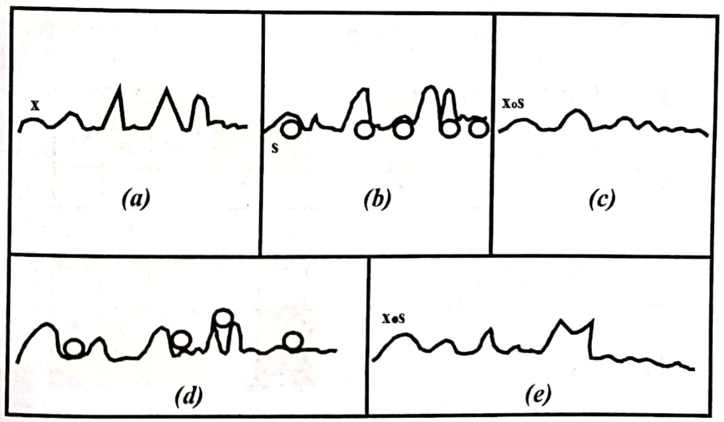


Fig. 1.26 Opening and Closing Gray-scale Image

surface is traversed. The opening x∘s then is the surface of the highest points reached by any part of the sphere as it slides over the entire undersurface of x. Fig. 1.26 (a) shows a scan line of gray-scale image as a continuous function to simplify. And the rolling ball in various positions as shown in fig. 1.26 (b). Fig. 1.26 (c) shows the complete result of opening x by s along the scan line. And last fig. 1.26 (d) and (e) shows the result of closing x by s.

Q.36. Give some application of gray-scale morphology.

Ans. Some various application of gray-scale morphology are as follows –

(i) Morphological Gradient – Dilation and erosion often are used to compute the morphological gradient of an image, denoted g.

$$g = (x \oplus \hat{s}) - (x \ominus s)$$

(ii) Textural Segmentation – A simple gray-scale image composed of two texture regions. The objective is to find the boundary between the two regions based on their textural content.

(iii) Granulometry – It is a field that deals principally with determining the size distribution of particles in an image.

(iv) Morphological Smoothing – One way to achieve smoothing is to perform a morphological opening followed by a closing.

(v) Top-hat Transformation – This transformation which owes its original name to the use of a cylindrical or parallelepiped structuring element function with a flat top is useful for enhancing detail in the presence of shading.

Q.37. Explain thinning operation of morphology.

Ans. Thinning a binary image down to a unit-width skeleton is useful not only to simplify the computational methods but also to decrease the pixels amount, needed for shape description. The thinning operation is based on the hit-or-miss transformation as given below –

$$X \otimes B = X - HM(X, B)$$

or

$$X \otimes B = X \cap HM(X, B)^c \quad \ldots(i)$$

Fig. 1.27 shows the different possibilities of structuring elements.

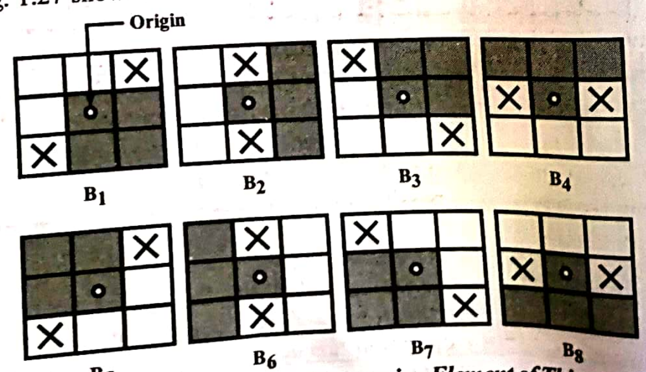


Fig. 1.27 Representation of Different Structuring Element of Thinning Operation

The operation of thinning is divided into two types – (i) It deletes boundary pixels of a connected component which are neither important for preserving the connectivity of image nor represent any significant geometrical features of an image. The operation converges if the connected skeleton does not change or vanishes even when the iteration operation continues. (ii) By a label showing the distance of the pixel to the boundary, thinning operation encodes distance information for every pixel of the pattern. The pixels set with local minimum distance labels is used to derive the resulting skeleton.

Q.38. Explain thickening operation of morphology.

Ans. Thickening is the morphological operation. It is used to grow selected regions of foreground pixels in binary images. This operation is somewhat such as dilation or closing. Following equation defines the thickening operation–

$$\overline{X} \cdot B = X \cup HM(X, B) \quad \ldots(i)$$

In equation (i), X represents the input image and B represents the structuring element. The thickened image comprises of the original image plus any additional foreground pixels switched on by hit or miss transform. The thickening operation is the dual of thinning, i.e., thinning the foreground is equivalent to thickening the background. This process is normally continued until it causes no further changes in the image. Fig. 1.28 shows the different structuring elements which can be used in the thickening operation.

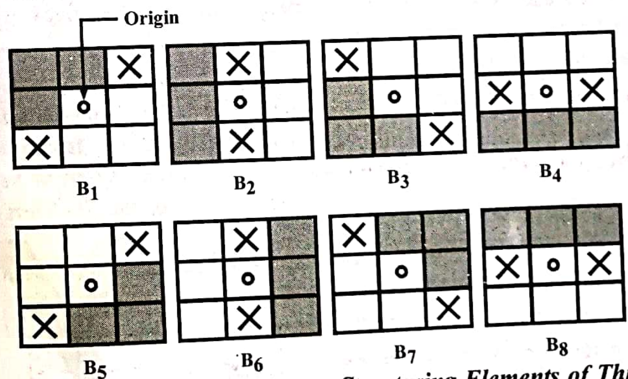


Fig. 1.28 Representation of Different Structuring Elements of Thickening Operation

The operation of thickening is computed by translating the origin of the structuring element to each possible pixel position in the image, and at each position comparing it with the underlying image pixels. When the black and white pixels in the structuring element exactly match the black and white pixels in the image, then the image pixel underneath the origin of the structuring element is set to black. Otherwise, it is remained unchanged means white. The applications of thickening operation are determining the convex hull and determining the skeleton by zone of influence.

Q.39. Explain about skeletons in morphology.

Ans. As represents in fig. 1.29, the notion of a skeleton, $S(A)$, of a set A is intuitively easy. We deduce from this figure that –

(i) When z represents a point of $S(A)$ and $(D)_z$ is the largest disk centered at z and contained in A, one cannot find a larger disk containing $(D)_z$ and involved in A. The disk $(D)_z$ is known as a maximum disk.

(ii) The boundary of A at two or more different locations is touched by the disk $(D)_z$.

In terms of erosions and openings, the skeleton of A is defined. That is, it can be represented that

$$S(A) = \bigcup_{k=0}^{K} S_k(A) \qquad \text{...(i)}$$

with $S_k(A) = (A \ominus kB) - (A \ominus kB) \ominus B$ ...(ii)

where B represents a structuring element, and $(A \ominus kB)$ represents k successive erosions of A, which are given below –

$$(A \ominus kB) = ((\ldots((A \ominus B) \ominus B) \ominus \ldots) \ominus B) \qquad \text{...(iii)}$$

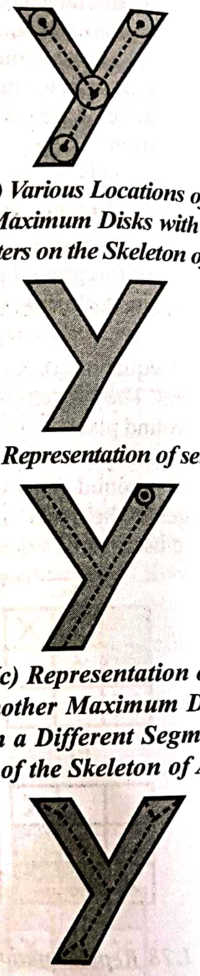
k times, and K is the last iterative step before A erodes to an empty set. In other words –

$$K = \max\{k \mid (A \ominus kB) \neq \phi\} \qquad \text{...(iv)}$$

The formulation provided in equations (i) and (ii) represents that $S(A)$ can be achieved as the union of the skeleton subsets $S_k(A)$. Also, it can be represented that A can be reconstructed from these subsets by using the equation–

$$A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB) \qquad \text{...(v)}$$

where $(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B) \oplus \ldots) \oplus B)$ ...(vi)



(a) Various Locations of Maximum Disks with Centers on the Skeleton of A

(b) Representation of set A

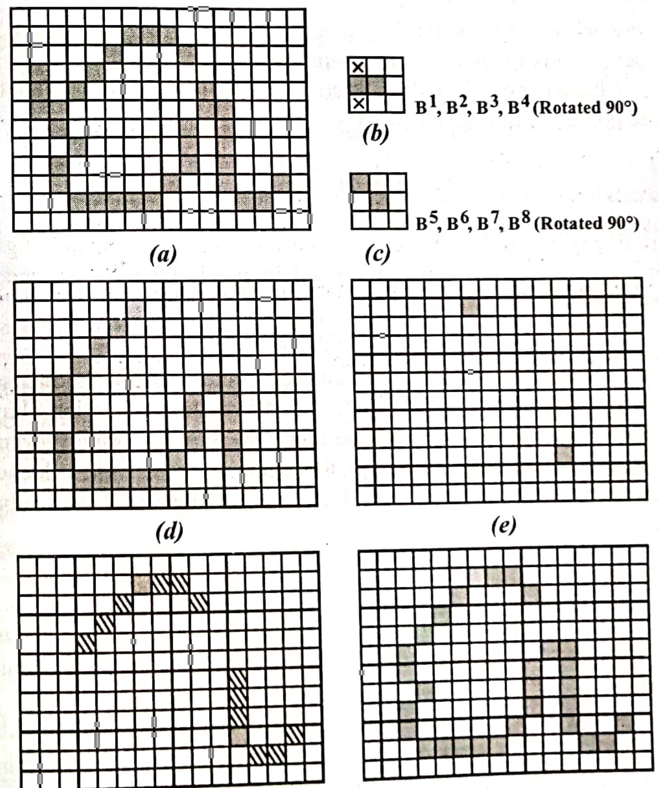
(c) Representation of another Maximum Disk on a Different Segment of the Skeleton of A

(d) Representation of Complete Skeleton

Fig. 1.29

Q.40. Explain about pruning in morphology.

Ans. Pruning procedures are an important complement to thinning and skeletonizing algorithms due to these methods tend to leave parasitic components that require to be "cleaned up" by postprocessing. In the automated recognition of hand-printed characters, a general method is to analyze the shape of the skeleton of each character. Spurs characterizes these skeletons. Spurs are caused during erosion by non uniformities in the strokes composing the characters. For handling this problem, a morphological method is developed. Beginning with the assumption that the length of a parasitic component does not exceed a specified number of pixels. The skeleton of a hand printed "a" is shown in fig. 1.30 (a). The parasitic component on the leftmost part of the character is illustrative of what we are interested in eliminating. By successively removing end point of parasitic branch, the solution is based on suppressing a parasitic branch. Of course, this also shortens other branches in the character but, in the absence of other structural information, the assumption in this example is that any branch with three or less pixels is to be removed. Thinning of an input set A with a order of structuring elements designed to detect only



(a) (b) (c) (d) (e) (f) (g)

Fig. 1.30 Representation of Pruning

end points obtains the desired result. That is, let

$$X_1 = A \otimes \{B\} \qquad ...(i)$$

where {B} represents the structuring element order which is represents in figs. 1.30 (b) and (c). The order of structuring elements comprises of two different structures, each of which is rotated 90° for a total of eight elements. The sign of × in fig. 1.30 (b) represents a "don't care" case, in the sense that it does not matter whether the pixel in that position has a value of 0 or 1. Various results reported in the literature on morphology are based on the use of a single structuring element, just like to the one in fig. 1.30 (b), but having "don't care" cases along the whole first column. This is not correct. For instance, this element would identify the point located in the eighth row, fourth column of fig. 1.30 (a) as an end point, hence, removing it and breaking connectivity in the stroke.

Performing equation (i) on A three times gives the set X_1 in fig. 1.30 (d). The next step is to "restore" the character to its original form, but with the parasitic branches eliminated. To do so first needs forming a set X_2 containing all end points in X_1 as given below –

$$X_2 = \bigcup_{k=1}^{8} (X_1 \otimes B^k) \qquad ...(ii)$$

where B^k represents the same end-point detectors as represented in figs. 1.30 (b) and (c). The next step using set A as a delimiter, is dilation of the end points three times –

$$X_3 = (X_2 \oplus H) \cap A \qquad ...(iii)$$

where, H represents a 3 × 3 structuring element of 1s and the intersection with A is performed after each step. Like in the condition of region filling and extraction of connected components, such conditional dilation prevents the creation of 1-valued elements outside the region of interest, like evidenced by the result represented in fig. 1.30 (f). Finally, the desired result can be obtained by the union of X_1 and X_3,

$$X_4 = X_1 \cup X_3 \qquad ...(iv)$$

in fig. 1.30 (g).

In more complex scenarios, equation (iii) use sometimes picks up the "tips" of some parasitic branches. This case can occur when the end points of these branches are near the skeleton. However, equation (i) may remove them due to they are valid points in A, they can be picked up again during dilation. Unless whole parasitic elements are picked up again, detecting and eliminating them is simple due to they are disconnected regions. At this juncture, a natural thought is that there must be simpler methods to solve this problem. For instance, we could just keep track of all deleted points and easily reconnect

the appropriate points to all end points remain after application of equation. This option is not invalid, but the advantage of the formulation just shown is that the use of easy morphological constructs solved the whole problem. In practical conditions, if a set of this type of tools is available, the advantage is that no new algorithm have to be written. The important morphological functions into a order of operations are easily combined.

Q.41. Explain the region growing technique.

Ans. A technique which groups pixels or subregions into larger regions based on predefined criteria for growth is known as region growing technique. The basic technique is to begin with a set of "seed" points and from these grow regions by appending to each seed those neighbouring pixels that have predefined properties similar to the seed. Choosing a set of one or more beginning points can be based on the problem nature. If a priori information is not available, the technique is to compute at every pixel the same set of properties which ultimately will be used to assign pixels to regions during the growing process. When the result of these computations represents values clusters, the pixels whose properties place them near the centroid of these clusters may be used as seeds.

The selection of similarity criteria depends not only on the problem under consideration, but also on the type of image data available. For example, the analysis of land-use satellite imagery depends heavily on the use of colour. To solve without the inherent information available in colour images, this problem would be significantly more difficult or even not possible. Region analysis must be carried out with a set of descriptors based on intensity levels and spatial properties, if the images are monochrome. When connectivity properties are not employed in the region-growing process, descriptors alone may yield misleading results. For example, a random arrangement of pixels is visualized with only three distinct intensity values. With the same intensity level, grouping pixels to form a "region" without paying attention to connectivity would yield a segmentation result which is meaningless.

The formulation of a stopping rule is the other problem in region growing. If no more pixels satisfy the criteria, region growth should stop for inclusion in that region. Criteria like intensity values, texture, and color are local in nature and do not take into account the history of region growth. Additional criteria which increase the power of region-growing algorithm utilize the concept of size, likeness between a candidate pixel and the pixels grown so far, and the shape of the region being grown. The use of these types of descriptors is based on the assumption that a model of expected results is at least partially available.

Q.42. Write a basic region-growing algorithm based on 8-connectivity.

Ans. Assume that f(x, y) represents an input image array. S(x, y) represents a seed array containing 1s at the locations of seed points and 0s elsewhere and Q represents a predicate to be applied at each location (x, y). Assume that an arrays f and S are of the same size. A basic region-growing algorithm based on 8-connectivity is given below –

(i) Start

(ii) Search all connected components in S(x, y).

(iii) Erode each connected component to one pixel. Label all this type of pixels found as 1. All other pixels in S are labeled 0.

(iv) Form an image f_Q such that, at a pair of coordinates (x, y), let $f_Q(x, y) = 1$ if the input image satisfies the given predicate, Q, at those coordinates and otherwise, let $f_Q(x, y) = 0$.

(v) Let g be an image formed by appending to each seed point in S all the 1-valued points in f_Q which are 8-connected to that seed point.

(vi) Label each connected component in g with a different region label. This is the segmented image achieved by region growing.

(vii) End.

Q.43. What do you mean by zooming and shrinking of digital images?
(R.G.P.V., Nov. 2019)

Ans. Image zooming is an important process in image processing. Basically zooming require two steps – the creation of new pixels locations and the assignment of gray level to those new locations. Suppose that we have an image of size 500*500 pixels and we want to enlarge it 1.5 times to 750*750 pixels. The spacing in the grid will be less than one pixel because we are fitting it over a small image. In order to perform gray level assignment for any point in the overlay, we lool for the closest pixels in the original image and assign its gray level to the new pixels in the grid. This method gray level assignment is called nearest neighbout interpolation.

Image shrinking is done in similar manner as described for zooming. The equivalent process of pixel replication is row and column deletion. For example, we want to shrink an image by one-half; we delete every row and column. How to present information effectively on small devices ? This is a main challenge for small-screen interface developers because viewing on a small screen is becoming more difficult in our daily lives. We must find effective ways to organize, show, and search data or results on small screen. One of the methods is to build zoom able users, and all need tools to enable them to control the zooming purposes.

UNIT 2

IMAGE REPRESENTATION AND DESCRIPTION – REPRESENTATION SCHEMES, BOUNDARY DESCRIPTORS, REGION DESCRIPTORS

Q.1. Write the various methods of representation. Discuss any one.

Ans. There are following methods of representation as given below –

(i) Boundary (border) following
(ii) Chain codes
(iii) Polygonal approximations using minimum perimeter polygons (MPP).
(iv) Signature
(v) Skeletons.

Boundary (Border) Following – By introducing a boundary following algorithms whose output is an ordered sequence of points. We consider –

(i) We are working with binary images where background and object points are labeled 0 and 1.

(ii) That images are padded with a border of 0s to avoid the possibility of an object merging with the image border.

The method is extended to multiple, disjoint regions by processing the regions individually.

Given a binary region R, the given boundary consists of the following steps –

(i) Consider the initial point b_0, be the uppermost, leftmost in the image which is labeled 1. Indicate by c_0 the west neighbour of b_0, c_0 is always a background point. Consider b_1 represent the first neighbour encountered whose value is 1 and c_1 is the point immediately proceeding b_1 in the sequence. Check the 8 neighbours of b_0, beginning at c_0 and proceeding in a clockwise direction. Store the locations of b_0 and b_1 for use in step (v).

(ii) Consider $b = b_1$ and $c = c_1$ as shown in fig. 2.1 (c).

(iii) Consider the 8-neighbours of b, beginning at c and proceeding in a clockwise direction, be represented by $n_1, n_2, \ldots, n_8$. Determine the first n_k labeled 1.

(iv) Consider $b = n_k$ and $c = n_{k-1}$.

(v) Repeat steps (iii) and (iv) until $b = b_0$ and the next boundary point computed is b_1.

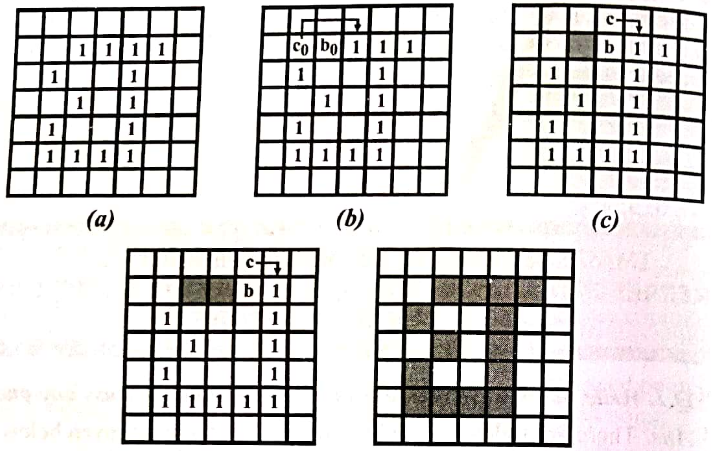
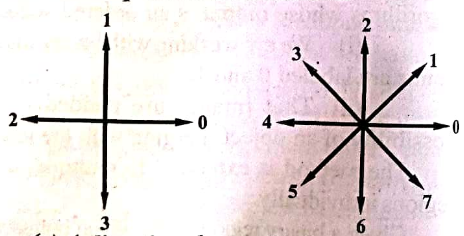


Fig. 2.1

Because n_k is the first 1-valued point determined in the clockwise scan, c is always a background point in step (iv). Sometimes, this algorithm is also called the Moore boundary tracking algorithm after move.

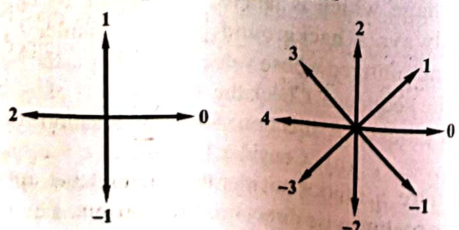
Q.2. Write short note on chain code representation method.

Ans. The chain code representation method was introduced in 1961 by Freeman. This representation is used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. Typically, chain code representation is based on 4 or 8 connectivity of the segments. The each segment direction is encoded by using a numbering method as shown in fig. 2.2. A boundary code formed as a sequence of such directional numbers is referred to as a Freeman chain code. The differential chain codes is shown in fig. 2.3.



(a) 4-directional ***(b) 8-directional***

Fig. 2.2 Chain Codes



(a) 4-directional ***(b) 8-directional***

Fig. 2.3 Differential Chain Codes

The chain code representation is constructed in following steps –

Step (i) – Choose a initial point of the curve. This point is represented by its absolute coordinates in the image.

Step (ii) – Every consecutive point is represented by a chain code showing the transition need to go from the present point to the next point on the curve.

Step (iii) – If the next point is the initial point then store the lengths of the curves into the file.

A variation of chain code is differential chain codes and differential chain code is denoted by K_i. Each differential chain code K_i is represented by the difference of the current chain code c_i and the preceding code c_{i-1} (i.e, $K_i = c_i - c_{i-1}$) There are two types of chain code –

(i) 4-directional chain code

(ii) 8-directional chain code.

Drawbacks of Chain Code – Any small disturbances along the boundary due to noise or imperfect segmentation cause change in the code that may not be related to the principal shape features of the boundary.

Q.3. Describe the polygonal approximations using minimum perimeter polygons.

Or

Write short note on polynomial approximation. (R.G.P.V., Nov. 2018)

Ans. A closed curve is approximated as a 2D polygon in case of polygonal approximation. The approximation becomes exact if the number of segments of the polygon is equal to the number of points in the boundary for a closed boundary. Thus each pair of adjacent point defines a segment of the polygon. The objective of a polygonal approximation is to capture the quality of the shape in a given boundary using a fewest possible number of segments. This approximation gives a simple representation of the planar object boundary.

Consider $X = \{x_1, x_2, ..., x_n\}$ is a set of points on the boundary of a planar object to be approximated using a polygon. It is defined as a partitioning of the set into N mutually exclusive and collectively exhaustive subsets $\lambda_1, \lambda_2, ..., \lambda_k$ such that each of the subsets may be approximated using a linear sequence. Polygonal approximation is obtained by minimization of an objective function in the form.

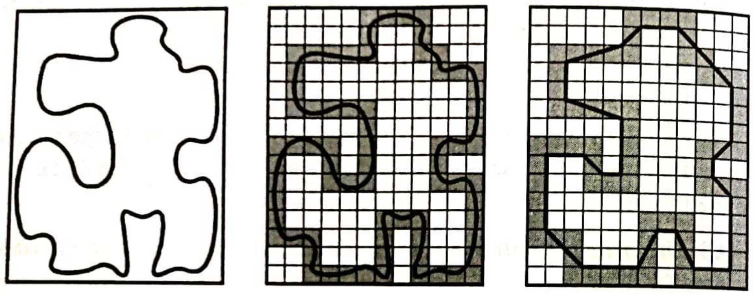
$$J = \sum_{i=1}^{n} d(x_i, l_j), x_i \in \lambda_j \qquad ...(i)$$

where l_j represents linear structure that approximates the points λ_j and d represents a measure of deviation. For image processing operations, approximation methods of modest complexity are well suited. One of the most powerful is representing a boundary by a minimum-perimeter polygon.

(MPP). It is explained in the following discussion –

(i) Foundation (ii) MPP algorithm.

(i) Foundation – An automatic appealing method for producing an algorithm to calculate MPPs is to enclose a boundary [see fig. 2.4 (a)] by a set of concatenated cells as illustrated in fig. 2.4 (b). Because it is permitted to shrink, the rubber band will be constrained by the inner and outer walls of the bounding region defined by the cells. The shape of a polygon of minimum perimeter is produced by this shrinking which circumscribes the region enclosed by the cell strip as shown in fig. 2.4 (c). In fig. 2.4 (c), all the vertices of the MPP coincide with corners of either the inner and/or the outer walls.



(a) (b) (c)

Fig. 2.4

(iii) MPP Algorithm – The set of cells enclosing a digital boundary is known as cellular complex. Consider the boundaries under consideration are not self intersecting, that leads to easily connected cellular complexes.

Letting write (W) and black (B) denote convex and mirrored concave vertices, respectively. The following observations are –

(a) The minimum perimeter polygon (MPP) bounded by a simply connected cellular complex, but it is not self intersecting.

(b) Each mirrored concave vertex of the minimum perimeter polygon is a black (B) vertex, but not every black (B) vertex of a boundary is a vertex of the minimum perimeter polygon.

(c) Each and every convex vertex of the minimum perimeter polygon is a white (W) vertex, but not every white (W) vertex of a boundary is a vertex of the minimum perimeter polygon.

(d) All white (W) vertices are inside the MPP and all black (B) vertices are outside the MPP.

(e) The uppermost, leftmost vertex in a sequence of vertices contained in a cellular complex, but it is always a white (W) vertex of the minimum perimeter polygon.

Q.4. Explain how merging scheme is used to solve the problem of polygonal approximation.

Ans. Merging schemes depend on average error or other criteria is used to solve the problem of polygonal approximation. Until the least square error line fit of the points merged so far exceeds a preset threshold, this method is used to merge points along a boundary. If this situation takes place, the error is set to zero, line parameters are collected and repeats the process, other point are merged, after all point has been merged, the intersections of adjacent line segments from the polygon vertices. Disadvantage of this method is that vertices in the final approximation do not always equivalent to inflections like corners to corners, because until the error threshold is increased, a next line is not begin.

Q.5. Explain in brief about splitting methods of polygonal approximation.

Ans. Splitting methods are used to subdivide a segment into two sections until a particular criterion is satisfied. For instance, a requirement might be that the maximum perpendicular length from a boundary segment to the line joining its two end points should not more than a preset threshold. When it does, the point containing greatest length from the line becomes a vertex, hence subdividing the initial segment into two different subsegments. Advantage of this method is seeking prominent inflection points. The best initial points are the two farthest points in the boundary for the closed region boundary.

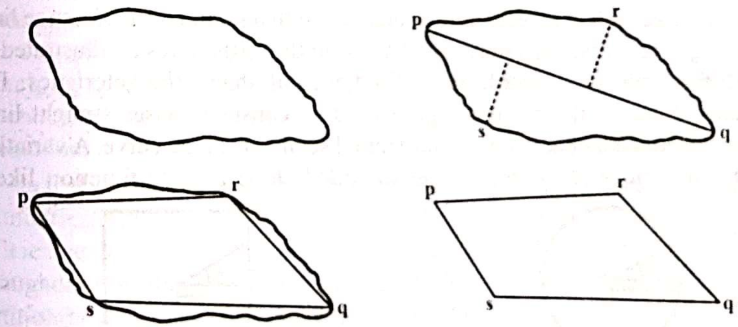


Fig. 2.5 Splitting Method

Q.6. Explain about signatures.

Ans. A one-dimensional functional representation of a boundary is called a signature. There are multiple methods to generate signature. The first one is to plot the distance from the centroid to the boundary as a function of angle. It does not matter that how a signature is generated. The objective is to minimize the boundary representation to a one dimension function which is simpler to explain than the original two dimensional boundary function. Method is used to generate the signatures are invariant to translation, but they do depend on

rotation and scaling. With respect to rotation, normalization may be obtained by searching a task to choice the same starting point to generate the signature without based on orientation of shape. There are several task to choice the same starting point. The first one is to choice the starting point as the point farthest from the centroid. The second is to choice the point on the eigen axis which is farthest from the centroid.

With respect to both axes, depend on the uniformity assumptions in scaling and which sampling is considered at equal intervals of θ, varies in size of shape output in varies in the equivalent signature values. To normalize for this is to scale all functions so that they span the same range of values. The main merit of this technique is simplicity. The disadvantage of this method is scaling of the whole function depends on maximum and minimum values. This dependence may be a cause of significant error from object to object when the shapes are noisy. A more rugged method is to divide each sample through the signature variance, let us assume that the value of variance is not equal to zero or so small which it generates computational difficulties. Variance use provides a variable scaling factor which is a inversely proportional to varies in size and works just like done by automatic gain control. Whatever the technique employed, the objective is to eliminate dependency on size while saving the basic shape of the waveforms. Distance versus angle is not the only method to generate a signature. The other method for generate the signature is to traverse the boundary and, equivalent to each point on the boundary, plot the angle between a line tangent to the boundary at that point and a reference line. Although the output signature different from the $r(\theta)$ curves as illustrated in fig. 2.6, would hold detail, about fundamental shape characteristics. For instance, because the tangent angle would be constant causes straight lines along the boundary equivalent to horizontal segments in the curve. A variation of this technique is to employ the so called slope density function like a

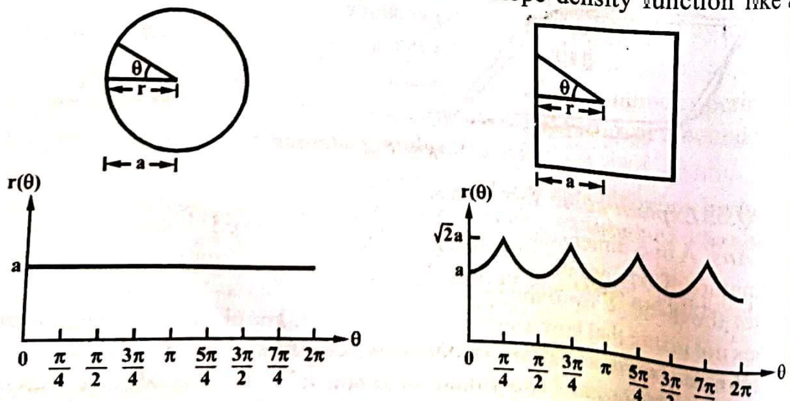


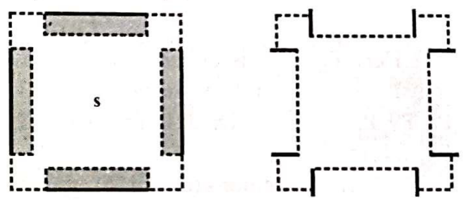
Fig. 2.6 Plot the Distance from the Centroid to the Boundary

signature. This function is referred to as a histogram of tangent angle values. The slope density function boundary sections which generating very quickly changing angles and reacts on strongly to the boundary sections with constant tangent angle because a histogram is a measure of values concentration.

Q.7. Discuss boundary segments.

Ans. Converting a small part of a boundary into segments is very useful. The boundary's complexity is reduced by decomposition and therefore, description process is simplified. If the boundary has one or more significant concavities which carry details of shape, this technique is attractive. For such a case, the convex hull of the region enclosed by the boundary is very useful for robust boundary decomposition. The convex hull H of an arbitrary set s is the smallest convex set having s. The set difference H-s is known as the convex deficiency D of the set s.

Fig. 2.7 represents a boundary segments by using this method. There are two object appearing in fig. 2.7 (a). The first one represents sets and its convex deficiency. The second one represent s boundary after partition. The boundary may be partitioned by the contour of s and marking the transition points which is meet into or out of a object of the convex deficiency. The output of this method is appearing in fig. 2.7 (b) in this case. In principle, this method is not dependent of boundary size and orientation.



(a) Set s ***(b) Boundary after Partition***

Fig. 2.7 Boundary Segments

In real life, due to noise, variations in segmentation and digitization, digital boundaries can be irregular. These effects appear in the form of convex deficiencies which include small, meaningless components scattered randomly throughout the boundary. A general method is to smooth a boundary prior to partitioning as compare to sort out these irregularities by postprocessing. There are several methods for this. The first one is traverse the boundary and change the coordinates of each pixel through the average coordinates of k of its neighbours along the boundary. This method works for small irregularities, but it is not easy to control and take more time to process. If value of k is small, it can be insufficient in some boundary segments while if value of k is large, it can provide extra smoothness.

Q.8. Write the algorithm to obtain the skeleton of the region.

Ans. There are two algorithms to obtain the skeleton of the region as given below –

Thinning Algorithm – Structural shape of a plane region can be reduce into a graph by obtaining the skeleton of the region via a thinning algorithm fo representation.

Thinning algorithm is as follows –

(i) If the given below condition are satisfied, then flags a contou point P1 for deletion.

(a) $2 \leq NZ(P1) \leq 6$ (b) $Z0(P1) = 1$

(c) $P2.P4.P6 = 0$ (d) $P4.P6.P8 = 0$

Here, $NZ(P1) = P2 + P3 + ... + P7 + P8 + P9$ and $NZ(P1)$ represents th number of non-zero neighbours of P1.

P9	P2	P3
P8	P1	P4
P7	P6	P5

(a)

0	0	1
1	P1	0
1	0	1

(b) NZ(P1) = 4 and Z0(P1) = 3

Fig. 2.8

Here P_i value is either 1 or 0 and is the number of zero to non-zer transitions in the ordered sequence P2, P3, P4, ..., P8, P9, P2. For example, $Z0(P1) = 3$ and $NZ(P1) = 4$ in fig. 2.8 (b).

(ii) Conditions (a) and (b) are not changed, while conditions (c) and (d) are changed into (c') and (d').

(c') $P2.P4.P8 = 0$

(d') $P2.P6.P8 = 0$.

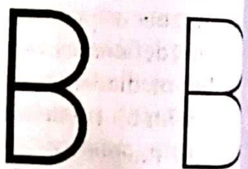


Fig. 2.9 Thinning

Skeleton Algorithms – The set of points whose distance from the neare boundary is locally maximum known as skeleton.

(i) Start

(ii) Transform of distance

$$u_k(m,n) = u_0(m,n) + \min_{\Delta(m,n;i,j)} \{u_{k-1}(i,j); (i,j) : \Delta(m,n;i,j) \leq 1\},$$

$$u_0(m,n) \triangleq u(m,n)$$

here $k = 1, 2, ...$

$\Delta(m, n; i, j)$ = Distance between (m, n) and (i, j)

k = Maximum thickness of the region, the transform is compute

(iii) The skeleton is the set of points –

$$\{(m,n) : u_k(m,n) \geq u_k(i,j), \Delta(m,n;i,j) \leq 1\}$$

(iv) End.

Q.9. What do you mean by the term skeleton ? ***(R.G.P.V., Nov. 2019)***

Ans. An important approach of representing the structural shape of a plane region is to reduce it to a graph. This reduction may be accomplished by obtaining the skeleton of the region via a thinning algorithm. A skeleton also called Skelectonization.

Skeletonization is a transformation of a component of a digital image into a subset of the original component. There are different categories of skeletonization methods – one category is based on distance transforms, and a specified subset of the transformed image is a distance skeleton. The original component can be reconstructed from the distance skeleton. Another category is defined by thinning approaches; and the result of skeletonization using thinning algorithms should be a connected set of digital curves or arcs. Motivations for interest in skeletonization algorithms are the need to compute a reduced amount of data or to simplify the shape of an object in order to find features for recognition algorithms and classifications. Additionally the transformation of a component into an image showing essential characteristics can eliminate local noise at the frontier.

Q.10. Explain some simple descriptors in boundary descriptors.

Ans. The length of a boundary represents one of its simplest descriptors. The number of pixels along a boundary provides a rough approximation of its length. The number of vertical and horizontal components plus $\sqrt{2}$ times the number of diagonal components provides its exact length for a chain coded curve with unit spacing in both directions.

The diameter of a boundary B is specified as given below –

$$\text{Diam}(B) = \max_{i,j}[D(p_i, p_j)] \quad \text{...(i)}$$

where D represents a distance measure and p_i and p_j are points on the boundary. The value of the diameter and the orientation of a line segment connecting the two extreme points which comprise the diameter are useful descriptors of a boundary. The minor axis of a boundary is described as the line perpendicular to the major axis, and of such length that a box passing via the outer four points of intersection of the boundary with the two axes completely encloses the boundary. The box just defined is known as the basic rectangle. The ratio of the major to the minor axis is known as the eccentricity of the boundary. This also is a useful descriptor.

Curvature is defined as the rate of change of slope. Usually, achieving reliable measures of curvature at a point in a digital boundary is not simple because these boundaries tend to be locally "ragged". Although, using the difference between the slopes of adjacent boundary segments as a descriptor of curvature at the point of intersection of the segments sometimes proves useful. Since the boundary is traversed in the clockwise direction, a vertex

point p is said to be part of a convex segment when the change in slope at p i nonnegative; otherwise, p is said to belong to a segment which is concave The curvature description at a point may be refined further by using limits i the change of slope. These descriptors must be used with care because the interpretation depends on the length of the individual segments relative to th overall length of the boundary.

Q.11. Explain in brief about shape numbers in boundary descriptors.

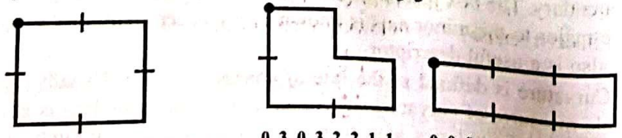
Ans. The shape number of boundary, based on the 4-directional code o fig. 2.2 (a), is expressed as the first difference of smallest magnitude. Th order n of a shape number is expressed as the number of digits in it representation of shape number. In addition, n is even for a closed boundar and value of n limits the number of possible different shapes.

Fig. 2.10 shows all the shapes of order 4, 6 and 8 along with their chain-cod representations, first differences, and corresponding shape numbers. By treatin the chain-code as a circular sequence, the first difference is calculated. Howeve the first difference of a chain-code is not dependent of rotation, usually the code boundary depends on the orientation of the grid. One method to normalize the gri orientation is by aligning the chain-code grid with the sides of the basic rectangl

Practically, for a desired shape order, the rectangle of order n is determine whose eccentricity best approximates that of the basic rectangle. This ne rectangle is used to establish the size of grid and procedure is used to achiev the chain code. The shape number follows from the first difference of thi code. However, the order of the resulting shape number equals n due to th way the grid spacing was chosen, boundaries with depressions comparable t this spacing sometimes yield shape numbers of order greater than n. In thi condition, a rectangle of order lower than n is specified and procedure i repeated until the resulting shape number is of order n.

Order 4 | Order 6

	Order 4	Order 6
Chain Code –	0 3 2 1	0 0 3 2 2 1
Difference –	3 3 3 3	3 0 3 3 0 3
Shape No. –	3 3 3 3	0 3 3 0 3 3



Chain Code –	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Difference –	3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0	3 0 0 3 3 0 0 3
Shape No. –	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3

Fig. 2.10 Representation of all Shapes of Order 4, 6 and 8

Q.12. Explain Fourier descriptors in boundary descriptors.

Ans. A K-point digital boundary in the xy-plane is represented in fig. 2.11. Beginning at an arbitrary point (x_0, y_0), coordinate pairs $(x_0, y_0), (x_1, y_1), (x_2, y_2), ..., (x_{K-1}, y_{K-1})$ are encountered in traversing the boundary in the counter-clockwise direction. These coordinates may be expressed in the form $x(k) = x_k$ and $y(k) = y_k$. With this notation, the boundary itself can be represented as the sequence of coordinates $s(k) = [x(k), y(k)]$, for $k = 0, 1, 2, ..., K-1$. Each coordinate pair may be treated as a complex number, hence,

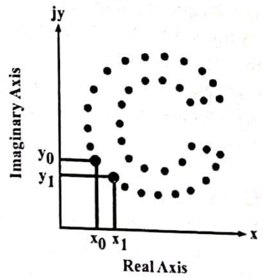


Fig. 2.11 A Digital Boundary and its Representation Like a Complex Sequence

$$s(k) = x(k) + jy(k) \qquad ...(i)$$

For $k = 0, 1, 2, ..., K-1$. That is, the x-axis is treated as the real axis and the y-axis as the imaginary axis of a sequence of complex numbers. However, the interpretation of the sequence was recast, the nature of the boundary itself was not changed. This representation has one great advantage that it decreases a 2-D to a 1-D problem.

The discrete Fourier transform of s(k) is given below –

$$a(u) = \sum_{k=0}^{K-1} s(k)e^{-j2\pi uk/K} \qquad ...(ii)$$

for $u = 0, 1, 2, ..., K-1$. The complex coefficients a(u) are known as the Fourier descriptors of the boundary. The inverse Fourier transform of these coefficients restores s(k) which is given below –

$$s(k) = \frac{1}{K}\sum_{u=0}^{K-1} a(u)e^{j2\pi uk/K} \qquad ...(iii)$$

for $k = 0, 1, 2, ..., K-1$. Assume that instead of all the Fourier coefficients, only the first P coefficients are employed. This is equivalent to setting $a(u) = 0$ for $u > P-1$ in equation (iii). The following approximation to s(k) is given below –

$$\hat{s}(k) = \frac{1}{P}\sum_{u=0}^{P-1} a(u)e^{j2\pi uk/P} \qquad ...(iv)$$

for $k = 0, 1, 2, ..., K-1$. However, to achieve each component of $\hat{s}(k)$, only P terms are employed, k still ranges from 0 to $K-1$. That is, the same number of points exists in the approximate boundary, but not as many terms are

employed in the reconstruction of each point. From the Fourier transform, the high-frequency components account for fine detail and global shape is determined by low-frequency components. Hence, the smaller P becomes, the more detail which is lost on the boundary. A few Fourier descriptors can be employed to capture the gross essence of a boundary. Due to these coefficients keep shape information, this property is valuable. Hence, they can be used as the basis for differentiating between distinct boundary shapes.

The descriptors should be as insensitive as possible to translation, rotation and scale changes. In conditions where results depend on the order in which points are processed, an additional constraint is that descriptors should be insensitive to the starting point. Fourier descriptors are not directly insensitive to these geometrical changes, but changes in these parameters can be related to easy transformations on the descriptors. For example, assume rotation, and recall from basic mathematical analysis that rotation of a point by an angle θ about the origin of the complex plane is accomplished by multiplying the point by $e^{j\theta}$. Doing so to every point of s(k) rotates the whole sequence about the origin. The rotated sequence is $s(k)\,e^{j\theta}$, whose Fourier descriptors are given below –

$$a_r(u) = \sum_{k=0}^{K-1} s(k)e^{j\theta}e^{-j2\pi uk/K}$$
$$= a(u)\,e^{j\theta} \quad ...(v$$

for u = 0, 1, 2, ..., K – 1. Hence, by a multiplicative constant term $e^{j\theta}$ rotation simply affects all coefficients equally.

Table 2.1 represents the Fourier descriptors for a boundary sequence s(k) which undergoes rotation, translation, scaling, and changes in starting point. The symbol Δ_{xy} is described as $\Delta_{xy} = \Delta x + j\Delta y$, hence, the notation $s_t(k) = s(k) + \Delta_{xy}$ denotes redefining the sequence as given below –

$$s_t(k) = [x(k) + \Delta x] + j[y(k) + \Delta y] \quad ...(vi$$

Table 2.1 Some basic Properties of Fourier Descriptors

Transformation	*Boundary*	*Fourier Descriptor*
Identity	$s(k)$	$a(u)$
Rotation	$s_r(k) = s(k)e^{j\theta}$	$a_r(u) = a(u)e^{j\theta}$
Translation	$s_t(k) = s(k) + \Delta_{xy}$	$a_t(u) = a(u) + \Delta_{xy}\delta(u)$
Scaling	$s_s(k) = \alpha s(k)$	$a_s(u) = \alpha a(u)$
Starting point	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2\pi k_0 u/K}$

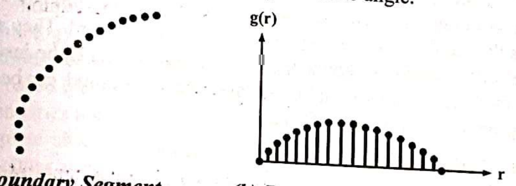
In other words, translation comprises of adding a constant displacement to all coordinates in the boundary. Translation has no effect on the descriptors except for u = 0, which include the impulse δ(u). Expression $s_p(k) = s(k - k_0)$ means redefining the sequence as given below –

$$s_p = x(k - k_0) + jy(k - k_0) \quad ...(vii)$$

which simply changes the starting point of the sequence to $k = k_0$ from k = 0. The last entry in table 2.1 represents that a change in starting point affects all descriptors in a different way, in the sense that the term multiplying a(u) depends on u.

Q.13. *Explain in brief about statistical moments in boundary descriptors.*

Ans. Statistical moments describe the shape of boundary segments like the mean, variance, and higher order moments. To see how this can be accomplished, consider fig. 2.12 (a), which represents the boundary segment, and fig. 2.12 (b), which represents the segment shown as a 1-D function g(r) of an arbitrary variable r. This function is achieved by connecting the two end points of the segment and rotating the line segment until it is horizontal. The coordinates of the points are rotated by the same angle.



(a) Boundary Segment ***(b) Representation as a 1-D Function***

Fig. 2.12

The amplitude of g can be treated as a discrete random variable v and form an amplitude histogram $p(v_i)$, i = 0, 1, 2, ..., A – 1, where A represents the number of discrete amplitude increments in which amplitude scale is divided. Then, keeping in mind that $p(v_i)$ is an estimate of the probability of value v_i occurring, the n^{th} moment of v about its mean is given below –

$$\mu_n(v) = \sum_{i=0}^{A-1}(v_i - m)^n p(v_i) \quad ...(i)$$

where

$$m = \sum_{i=0}^{A-1} v_i p(v_i) \quad ...(ii)$$

The quantity m is recognized as the mean or average value of v and μ_2 is recognized as its variance. Normally, only the first few moments are needed to differentiate between signatures of clearly distinct shapes.

An alternative method is to normalize g(r) to unit area and treat it as a histogram. In other words, $g(r_i)$ is treated as the probability of value r_i occurring.

In this condition, r is treated as the random variable and the moments are

$$\mu_n(r) = \sum_{i=0}^{K-1} (r_i - m)^n \; g(r_i) \qquad ...(iii)$$

where

$$m = \sum_{i=0}^{K-1} r_i \; g(r_i) \qquad ...(iv)$$

In this notation, K is the number of points on the boundary and $\mu_n(r)$ is directly related to the shape of g(r). For example, the spread of the curve about the mean value of r is measured by the second moment $\mu_2(r)$ and third moment $\mu_3(r)$ measures its symmetry with reference to the mean.

Generally, what we have accomplished is to decrease the description task to that of describing 1-D functions. However moments are by far the most popular procedure, they are not the only descriptors employed for this purpose. For instance, another procedure involves computing the 1-D discrete Fourier transform, achieving its spectrum and using the first q components of the spectrum to define g(r). The advantage of moments as compared to other method is that implementation of moments is straightforward. They also keep a physical interpretation of boundary shape. From fig. 2.12, the insensitivity of this method to rotation is clear. Size normalization, if desired, can be obtained by scaling the range of values of g and r.

Q.14. Explain in brief about shape descriptors.

Ans. Shape descriptors are a powerful tool used in wide spectrum of computer vision and image processing tasks like object matching, classification, recognition and identification. Many approaches have been developed. There are a number of generic shape descriptors that are capable of providing a high dimensionality feature vector that accurately describes specific shapes (for example, Fourier descriptors and moment invariants). Alternatively, other descriptors describe some single characteristic that is present over a variety of shapes, like circularity, ellipticity, rectangularity, triangularity, rectilinearity, complexity, mean curvature, symmetry, etc. Even for a single characteristic of shapes there often exist many alternative measures which are sensitive to different aspects of the shape. Very likely, the shape convexity is a shape property with the largest number of different methods defined for its evaluation. The need for alternative measures is caused by the fact that there is no a single shape descriptor which is expected to perform efficiently in all possible applications.

Generally speaking, there are two approaches to analyze shapes – boundary based (which use the information from boundary points only) and area based ones (which use all the shape points). It could be said that, in the past, more attention has been given to the area based methods. The area based methods are more robust (e.g. with respect to noise). Although not mentioned often, an

additional reason for a larger number of methods that are based on 'interior' shape points, rather than methods based on boundary points, is that area based methods are usually simpler to compute. For example, to estimate accurately the area of a given shape, it is sufficient to enumerate the number of pixels inside the shape, while the perimeter estimation is not a straightforward task. Depending on particular situation and conditions assumed different methods have to be used.

Another example would be geometric (area) moment invariants, these are easily and accurately computable from the corresponding object image, while their boundary based analogues involve computation of path integrals, which are not simple to be estimated from discrete data, which are mainly used in image processing and computer vision applications. On the other side, the boundary based methods are more suitable for a high precision computer vision and image processing tasks. They are able to cope much easier with objects with partially extracted boundaries or with partially occluded objects. Robustness is a very desirable property when we work with low quality data (e.g., noisy images or low resolution images), but recently, due to progress in image technology, high quality data can be provided, and the use of boundary based methods becomes highly acceptable in many applications. In addition, boundary based methods could have a much lower time complexity because shape boundaries are represented by a significantly smaller number of pixels than complete shapes are. Of course, there are methods which cannot be classified either as boundary based or volume (area) based ones. For example, a very popular shape measure, the shape compactness.

$$C_{st}(S) = \frac{4.\pi.\text{Area of S}}{(\text{Perimeter of S})^2}$$

obviously uses both boundary and interior information. This quantity indicates how much a given shape differs from a perfect circular disc, which is understood as the most compact shape. Accordingly, the highest possible compactness (equal to 1) is assigned to circular disc. Finally, there are methods which use only information from specific points or specific boundary parts (e.g., parts belonging to the convex hull of the shape considered).

Here, we focus on shape analysis techniques based on the use of a set of suitably selected shape descriptors/measures. Generally speaking, a shape measure is a quantity which relates to a particular shape characteristic. More formally, a certain shape measure D(S) (related to a certain shape descriptor) maps a given planar shape S into a real number. In order to be applicable in object classification, recognition or identification task, any shape measure is expected to be invariant with respect to similarity transformations (translation, rotation, and scaling). Also, shape measures are preferred to be given in a normalized form. An easiest way to achieve a normalized form is to apply a

scaling transformation which would preserve that D(S) varies through th interval [0, 1] [or even better through the interval (0, 1)] while S varies throug the set of bounded compact planar regions.

Thus, common desirable properties of a given shape measure D(S) are

(i) $D(S) \in [0, 1]$

(ii) $D(S) = 1$

if S satisfies a certain property (here called a shape descripto for which, actually, the shape measure D(S) is designed.

(iii) D(S) is invariant with respect to the similarity transformation

(iv) For any $\delta > 0$ there is a shape S such that $D(S) < \delta$ (e.g., 0 i the best possible lower bound for D(S)).

Q.15. Explain in brief about some simple descriptors in regional descriptor

Ans. A region area is described as the number of pixels in the region. Th perimeter of a region represents the length of its boundary. However, perimete and area are sometimes employed as descriptors, they apply primarily to condition in which the size of the regions of interest is invariant. A more frequent use o these two descriptors is in measuring compactness of a region, described a (perimeter)^2/area. A slightly different descriptor of compactness is the circulari ratio, described as the ratio of the area of a region to the area of a circle havin the same perimeter. $P^2/4\pi$ is the area of a circle with perimeter length P. Henc the circularity ratio R_c is obtained by the equation, which is given below –

$$R_c = \frac{4\pi A}{P^2} \quad ...(i$$

where, A represents the area of the region in question and P represents th length of its perimeter. For a circular region, the value of this measure is 1 an for square region, the value of this measure is $\pi/4$. Compactness is dimensionless measure. Hence, compactness is insensitive to uniform scal changes ; it is insensitive also to orientation, ignoring, of course, computationa errors which can be introduced in resizing and rotating a digital region. Anothe easy measures employed like region descriptors include the mean and media of the intensity levels, the minimum and maximum intensity values, and th number of pixels with values above and below the mean.

Q.16. Explain in brief about topological descriptors in regional descriptors

Ans. Topological properties are useful for global descriptions of region in the image plane. Topology is the study of properties of a figure which ar unaffected by any deformation, as long as there is no tearing or joining of th figure. For example, a region with two holes is shown in fig. 2.13. Hence when a topological descriptor is described by the number of holes in th region, this property obviously will not be affected by a stretching or rotatio

transformation. Usually, when the region is turn or folded, then the number of holes will change. As stretching affects distance, topological properties do not depend on the notion of distance or any properties implicitly based on the concept of a distance measure.

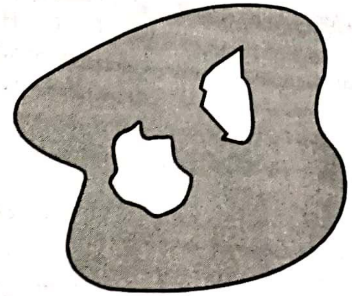


Fig. 2.13 Representation of a Region with Two Holes

For region description, another topological property useful is the number of connected components. A region with three connected components is represented in fig. 2.14.

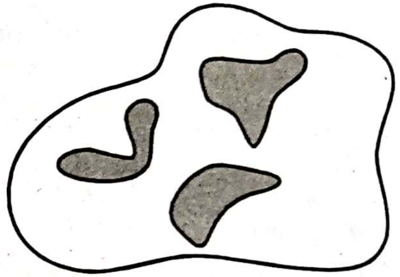


Fig. 2.14 Representation of a Region with Three Connected Components

In a figure, connected components C and the number of holes H can be used to describe the Euler number E –

$$E = C - H \quad ...(i)$$

The Euler number is also a topological property. Fig. 2.15 represents the regions, for example, have Euler numbers equal to 0 and –1, respectively, due to the "A" has one connected component and one hole and the "B" one connected component but two holes. Regions shown by straight-line segments (known as polygonal networks), include a particularly easy interpretation in terms of the Euler number. A polygonal network is represented by fig. 2.16. Classifying interior regions of this type of network into faces and holes is often important. Representating the number of vertices by "V", the number of edges by "Q", and the number of faces by F gives the following relationship, known as

Fig. 2.15 Representation of Regions with Euler Numbers Equal to 0 and –1

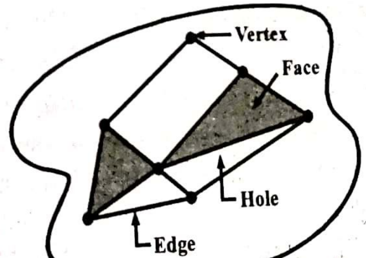


Fig. 2.16 A Polygonal Network is Contained by Region

Euler formula which is given below –

$$V - Q + F = C - H$$

which, in view of equation (i), is equal to the Euler number – ...(ii

$$V - Q + F = C - H = E$$ ...(iii

The network in fig. 2.16 has one connected region, three holes, seven vertices, eleven edges, two faces. Hence, the Euler number is –2.

$$7 - 11 + 2 = 1 - 3 = -2$$

An additional feature which is useful in characterizing regions in a scene is given by topological descriptors.

Q.17. Write short note on texture in regional descriptors.

Ans. An important method to region description is to quantify its texture content. However no formal definition of texture presents, intuitively this descriptor provides measures of properties like smoothness, coarseness, and regularity. Statistical, structural and spectral are the three principal methods employed in image processing to describe the texture of a region. Statistical methods yield characterizations of textures like smooth, coarse, grainy, and so on. Structural methods deal with the arrangement of image primitives like the description of texture based on regularly spaced parallel lines. Spectral methods are based on properties of the Fourier spectrum. Spectral methods are used primarily to detect global periodicity in an image by identifying high-energy, narrow peaks in the spectrum.

Q.18. Derive the expression of moment invariants in regional descriptors.

Ans. The 2-D moment of order (p + q) of a digital image f(x, y) of size M × N is given below –

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y)$$ ...(i)

where, p = 0, 1, 2, ... and q = 0, 1, 2, ... are integers. The corresponding central moment of order (p + q) is given below –

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$ ...(ii)

for p = 0, 1, 2, ... and q = 0, 1, 2, ...

where $$\bar{x} = \frac{m_{10}}{m_{00}} \text{ and } \bar{y} = \frac{m_{01}}{m_{00}}$$ ...(iii)

The normalized central moments, denoted η_{pq}, are given below –

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$$ ...(iv)

where $$\gamma = \frac{p+q}{2} + 1$$ ...(v)

for p + q = 2, 3, ...

From the second and third moments, a set of seven invariant moments can be derived –

$$\phi_1 = \eta_{20} + \eta_{02}$$ ...(vi)

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$ ...(vii)

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$ ...(viii)

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$ ...(ix)

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$ ...(x)

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$ ...(xi)

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$ ...(xii)

This set of moments is invariant to translation, scale change, mirroring and rotation.

Q.19. Explain the use of principal components for description.

Ans. Assume that we are given the three component images of such a colour image. The three image can be treated as a unit by expressing each group of three corresponding pixels as a vector.

For example, let a_1, a_2 and a_3 respectively, be the values of pixel in each of the three RGB component images. These three elements can be expressed in the form of a 3-D column vector, **a** where

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Here, this vector represents one common pixel in all three images. If the images are of size M × N, there will be a total of K = MN 3D vectors after all the pixels are represented in this manner. If we have n registered images, the vectors will be n-dimensional

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$ ...(i)

The assumption is that all vectors are column vectors (i.e. matrices of order n × 1). We can write them on a line of text simply by expressing them as $\mathbf{a} = (a_1, a_2, ..., a_n)^T$, where "T" is transpose.

We can treat the vectors as random quantities, just like we did wh[…] constructing an intensity histogram. The only difference is that, instead[…] talking about quantities like the mean and variance of the random variables,[…] now talk about mean vectors and convariance matrices of the random vect[…] The mean vector of the population is defined as –

$$M_a = E(a)$$

Here, E{.} is the expected value of the argument, and the subscript deno[…] that **M** is associated with the population of **a** vectors. Recall the expec[…] value of a vector or matrix is obtained by taking the expected value of ea[…] element. The covariance matrix of the vector population is defined as

$$Q_a = E\{(a - M_a)(a - M_a)^T\}$$

Here, **a** is n dimensional, Q_a and $(a - M_a)(a - M_a)^T$ are matrices of ord[…] n × n. Element q_{ii} of Q_a is the variance of a_i, the i^{th} component of the vectors in the population, and element q_{ij} of Q_a is the convariance betwe[…] element a_i and a_j of these vectors. The matrix Q_a is real and symmetric. elements a_i and a_j are uncorrelated, their covariance is zero and, therefore $= q_{ji} = 0$ all these definitions reduce to their familiar one dimensional counterpa[…] when n = 1.

BINARY MACHINE VISION – THRESHOLDING, SEGMENTATION, CONNECTED COMPONENT LABELING, HIERARCHICAL SEGMENTATION, SPATIAL CLUSTERING, SPLIT & MERGE, RULE BASED SEGMENTATION, MOTION-BASED SEGMENTATION

Q.20. What do you mean by thresholding ?

Ans. A binary image is obtained using an appropriate segmentation of[…] gray-scale image. If the intensity values of an object are in an interval and t[…] intensity values of the backgrounds pixels are outside this interval, a bina[…] image can be obtained using a thresholding operation that sets the points i[…] that interval to 1 and points outside that range to 0. Thus, for binary visio[…] segmentation and thresholding are synonymous. Many cameras have bee[…] designed to perform this thresholding operation in hardware. The output […] such a camera is a binary image. In most applications, however, cameras giv[…] a gray-scale image and the binary image is obtained using thresholding.

Thresholding is a method to convert a gray-scale image into a binar[…] image so that objects of interest are separated from the background. Fo[…] thresholding to be effective in object-background separation, it is necessar[…] that the objects and background have sufficient contrast and that we kno[…]

the intensity levels of either the objects or the background. In a fixed thresholding scheme, these intensity characteristics determine the value of the threshold.

Let us assume that a binary image B[x, y] is the same as a thresholded gray image $G_T[x, y]$ which is obtained using a threshold T for the original gray image G[x, y]. Thus

$$B[x, y] = G_T[x, y]$$

where for a darker object on a lighter background

$$G_T[x, y] = \begin{cases} 1 & \text{if } G[x, y] \le T \\ 0 & \text{otherwise} \end{cases}$$

If it is known that the object intensity values are in a range $[T_1, T_2]$, then we may use

$$G_T[i, j] = \begin{cases} 1 & \text{if } T_1 \le G[x, y] \le T_2 \\ 0 & \text{otherwise} \end{cases}$$

A general thresholding scheme in which the intensity levels for an object may come from several disjoint intervals may be represented as

$$G_T[i, j] = \begin{cases} 1 & \text{if } G[x, y] \in X \\ 0 & \text{otherwise} \end{cases}$$

where X is a set of intensity values for object components.

Q.21. Write the algorithm to estimate the thresholding values.
(R.G.P.V., June 2015)

Ans. The algorithm to estimate the thresholding values is as follows –

(i) Start

(ii) Select an initial estimate for T(threshold).

(iii) Segment the image using T. This will generate two sets of pixels – the first one is G_1 which holds all pixels with gray level values > T and the second one is G_2 which holds pixels with values ≤ T.

(iv) For the pixels in regions G_1 and G_2, calculate the average gray level values μ_1 and μ_2.

(v) Calculate a new threshold value –

$$T = \frac{\mu_1 + \mu_2}{2}$$

(vi) Until the difference in T in successive iterations is smaller than a predefined parameter T_0, repeat steps from (iii) to (v).

(vii) End.

Q.22. Explain global and adaptive thresholding techniques.
(R.G.P.V., June 2015)

Ans. **Global Thresholding Technique** – It is simplest and most widely used of all possible segmentation techniques. In this method, a threshold value

of θ is selected and given condition is imposed –

$$x(k, l) = \begin{cases} 1 & \text{if } x(k, l) \geq 0 \\ 0 & \text{else} \end{cases} \quad ...(i)$$

Equation (i) represents a full description of a binarisation algorithm. It does not define that how to choose the threshold parameter θ. The value of θ has to be chosen in an optimal way. If pixels from different segments overlap in their use of intensities global thresholding will be affected. A method like minimum-error may estimate the underlying cluster parameters and select the thresholds to minimize the classification error when error is due to noise. Variable thresholding can be used when the overlap is due to variation in illumination across the image. Hence, this may be considered as a form of local segmentation.

In this method, using image intensity value, a gray scale image is transformed into a binary image. All pixels values are greater than the global threshold values are indicated by one and remaining are indicated by zero.

Adaptive Thresholding Technique – In this method, the thresholding operations are based on local image features. In the condition of poorly illuminated images, local thresholding is more useful as compared to global thresholding.

A perfectly segmentable histogram can be converted into a histogram which cannot be partitioned effectively through a single global threshold by an uneven illumination. This problem can be solved by dividing the original image into subimages. After that, to segment each subimage utilize a different threshold. Two problems associated with this approach are as follows – the first one is how to subdivide the image and the second one is how to estimate the threshold for each resulting subimage. Because the threshold used for each pixel depends on the pixel position with respect to subimages, this type of thresholding is known as adaptive thresholding.

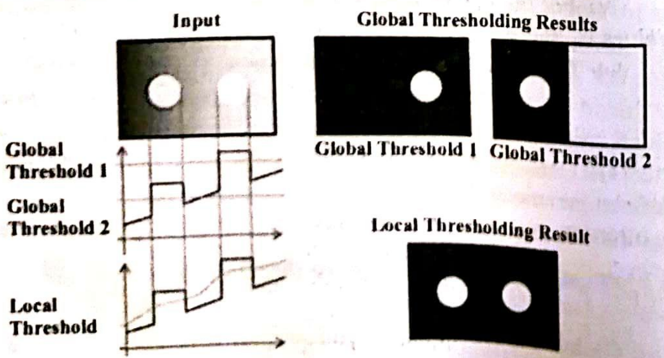


Fig. 2.17 Representation of Global and Local (Adaptive) Thresholding

Q.23. What is image segmentation ? Explain.

Ans. Image segmentation subdivides an image into its constituent objects. The level to which the subdivision is carried depends on the problem to be solved, i.e. segmentation should stop if the objects of interest in an application have been isolated. For instance, in the automated analysis of electronic assemblies interest lies in analyzing products' images to find the presence or absence of specific anomalies like broken connection paths or missing components. There is no need to carry segmentation past the level of detail required to identify those elements. Segmentation of nontrivial images is very difficult task. The eventual success or failure of computerized analysis methods is determined by accuracy of segmentation. Considerable care is therefore, required to improve the probability of rugged segmentation. Image segmentation is used to extract multiple features of the image that may split or merged in manner to develop objects of interest. Analysis and interpretation can be performed these objects.

The goal of image segmentation is to divide an image into several parts/segments having similar features or attributes. The basic applications of image segmentation are – Content-based image retrieval, medical imaging, object detection and recognition tasks, automatic traffic control systems and video surveillance, etc. The image segmentation can be classified into two basic types – Local segmentation (concerned with specific part or region of image) and global segmentation (concerned with segmenting the whole image, consisting of large number of pixels). The image segmentation approaches can be categorized into two types based on properties of image.

(i) Discontinuity Detection Based Approach – This is the approach in which an image is segmented into regions based on discontinuity. The edge detection based segmentation falls in this category in which edges formed due to intensity discontinuity are detected and linked to form boundaries of regions.

(ii) Similarity Detection Based Approach – This is the approach in which an image is segmented into regions based on similarity. The techniques that falls under this approach are – thresholding techniques, region growing techniques and region splitting and merging. These all divide the image into regions having similar set of pixels. The clustering techniques also use this methodology. These divide the image into set of clusters having similar features based on some predefined criteria.

In other words, also we can say that image segmentation can be approached from three perspectives – Region approach, edge approach and data clustering. The region approach falls under similarity detection and edge detection and boundary detection falls under discontinuity detection. Clustering techniques are also under similarity detection.

Q.24. Explan in detail the classification of image segmentation techniques.

Ans. The popular techniques used for image segmentation are – thresholding method, edge detection based techniques, region based techniques, clustering based techniques, watershed based techniques, partial differential equation based and artificial neural network based techniques etc. These all techniques are different from each other with respect to the method used by these for segmentation.

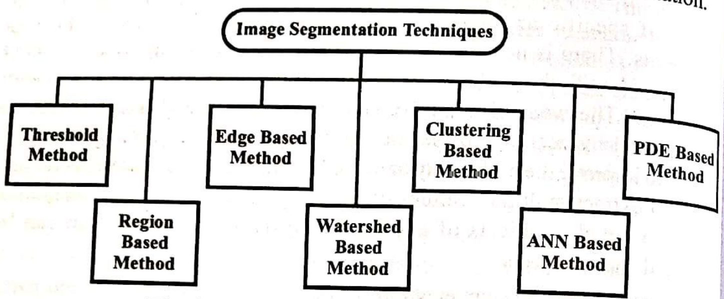


Fig. 2.18 Image Segmentation Techniques

(i) Thresholding Methods – These are the simplest methods for image segmentation. These methods divide the image pixels with respect to their intensity level. These methods are used over images having lighter objects than background. The selection of these methods can be manual or automatic i.e., can be based on prior knowledge or information of image features. There are basically three types of thresholding –

(a) Global Thresholding – This is done by using any appropriate threshold value/T. This value of T will be constant for whole image. On the basis of T the output image q(x, y) can be obtained from original image p(x, y) as –

$$q(x, y) = \begin{cases} 1, & \text{if } p(x, y) > T \\ 0, & \text{if } p(x, y) \le T \end{cases}$$

(b) Variable Thresholding – In this type of thresholding, the value of T can vary over the image. This can further be of two types –

(1) Local Threshold – In this case the value of T depends upon the neighbourhood of x and y.

(2) Adaptive Threshold – The value of T is a function of x and y.

(c) Multiple Thresholding – In this type of thresholding, there are multiple threshold values like T0 and T1. By using these output image can be computed as –

$$q(x, y) = \begin{cases} m, & \text{if } p(x, y) > T1 \\ n, & \text{if } p(x, y) \le T1 \\ o, & \text{if } p(x, y) \le T0 \end{cases}$$

The values of thresholds can be computed with the help of the peaks of the image histograms. Simple algorithms can also be generated to compute these.

(ii) Edge Based Segmentation Methods – These techniques are well developed techniques of image processing on their own. The edge based segmentation methods are based on the rapid change of intensity value in an image because a single intensity value does not provide good information about edges. Edge detection techniques locate the edges where either the first derivative of intensity is greater than a particular threshold or the second derivative has zero crossings. In edge based segmentation methods, first of all the edges are detected and then are connected together to form the object boundaries to segment the required regions. The basic two edge based segmentation methods are – Gray histograms and Gradient based methods. To detect the edges one of the basic edge detection techniques like sobel operator, canny operator and Robert's operator etc. can be used. Result of these methods is basically a binary image. These are the structural techniques based on discontinuity detection.

(iii) Region Based Segmentation Methods – These methods segments the image into various regions having similar characteristics. There are two basic techniques based on this method –

(a) Region Growing Methods – The region growing based segmentation methods are the methods that segments the image into various regions based on the growing of seeds (initial pixels). These seeds can be selected manually (based on prior knowledge) or automatically (based on particular application). Then the growing of seeds is controlled by connectivity between pixels and with the help of the prior knowledge of problem, this can be stopped. Let p(x, y) be the original image that is to be segmented and s(x, y) is the binary image where the seeds are located. Also let, 'T' be any predicate which is to be tested for each (x, y) location. Then basic algorithm (based on 8-connectivity) steps for region growing method are –

(1) First of all, all the connected components of 's' are eroded.

(2) Compute a binary image P_T, where $P_T(x, y) = 1$, if $T(x, y)$ = True.

(3) Compute a binary image 'q', where q(x, y) = 1, if $P_T(x, y) = 1$ and (x, y) is 8-connected to seed in 's'.

These connected components in 'q' are segmented regions.

(b) Region Splitting and Merging Methods – The region splitting and merging based segmentation methods use two basic techniques i.e., splitting and merging for segmenting an image into various regions. Splitting

stands for iteratively dividing an image into regions having similar characteristic and merging contributes to combining the adjacent similar regions. Following diagram shows the division based on quad tree. The basic algorithm steps for region splitting and merging are –

(1) First of all the R_1 is equal to P.

(2) Each region is divided into quadrants for which $T(R_i)$ = False.

(3) If for every region, $T(R_j)$ = True, then merge adjacent regions R_i and R_j such that $T(R_i \cup R_j)$ = True.

(4) Repeat step 3 until merging is impossible.

where 'p' be the original image and 'T' be the particular predicate.

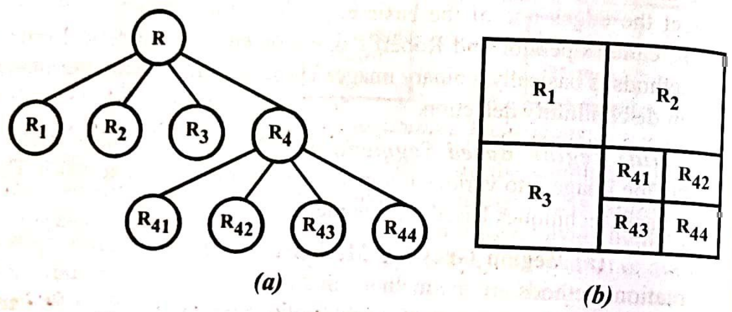


Fig. 2.19 Division of Regions Based on Quad Tree

(iv) Clustering Based Segmentation Method – The clustering based techniques are the techniques, which segment the image into clusters having pixels with similar characteristics. Data clustering is the method that divides the data elements into clusters such that elements in same cluster are more similar to each other than others. There are two basic categories of clustering methods – hierarchical method and partition based method. The hierarchical methods are based on the concept of trees. In this the root of the tree represents the whole database and the internal nodes represent the clusters. On the other side the partition based methods use optimization methods iteratively to minimize an objective function. In between these two methods there are various algorithms to find clusters. There are basic two types of clustering –

(a) Hard Clustering – This is a simple clustering technique that divides the image into set of clusters such that one pixel can only belong to only one cluster. In other words it can be said that each pixel can belong to exactly one cluster. These methods use membership functions having values either 1 or 0 i.e., one either certain pixel can belong to particular cluster or not. An example of a hard clustering based technique is one k-means clustering based technique known as HCM. In this technique, first of all the centers are computed then each pixel is assigned to nearest center. It emphasizes on maximizing the intra cluster similarity and also minimizing the inter cluster equality.

(b) Soft Clustering – This is more natural type of clustering because in real life exact division is not possible due to the presence of noise. Thus soft clustering techniques are most useful for image segmentation in which division is not strict. The example of such type of technique is fuzzy c-means clustering. In this technique pixels are partitioned into clusters based on partial membership i.e., one pixel can belong to more than one clusters and this degree of belonging is described by membership values. This technique is more flexible than other techniques.

(v) Watershed Based Methods – These methods uses the concept of topological interpretation. In this the intensity represents the basins having hole in its minima from where the water spills. When water reaches the border of basin the adjacent basins are merged together. To maintain separation between basins dams are required and are the borders of region of segmentation. These dams are constructed using dilation. The watershed methods consider the gradient of image as topographic surface. The pixels having more gradient are represented as boundaries which are continuous.

(vi) Partial Differential Equation Based Segmentation Method – The partial differential equation based methods are the fast methods of segmentation. These are appropriate for time critical applications. There are basic two PDE methods – non-linear isotropic diffusion filter (used to enhance the edges) and convex non-quadratic variation restoration (used to remove noise). The results of the PDE method is blurred edges and boundaries that can be shifted by using close operators. The fourth order PDE method is used to reduce the noise from image and the second other PDE method is used to better detect the edges and boundaries.

(vii) Artificial Neural Network Based Segmentation Method – The ANN based segmentation methods simulate the learning strategies of human brain for the purpose of decision making. At present this method is mostly used for the segmentation of medical images. It is used to separate the required image from background. A neural network is made of large number of connected nodes and each connection has a particular weight. This method is independent of PDE. In this method the problem is converted to issues which are solved using neural network. This method has basic two steps – extracting features and segmentation by neural network.

Q.25. Give comparison of various segmentation techniques.

Ans. The comparison of various segmentation techniques is given in tabl 2.2.

Table 2.2 Comparison of Various Segmentation Techniques

S. No.	Segmentation Technique	Description	Advantages	Disadvantages
(i)	Thresholding method	This method is based on the histogram peaks of the image to find particular threshold values.	No need of previous information, simplest method	This method, highl dependent on peaks spatial details are no considered
(ii)	Edge based method	This method is based on discontinuity detection.	Good for images having better contrast between objects.	Not suitable for wrong detected or too many edges.
(iii)	Region based method	This method is based on partitioning image into homogeneous regions.	More immune to noise, useful when it is easy to define similarity criteria.	Expensive method in terms of time and memory.
(iv)	Clustering method	This method is based on division into homogeneous clusters.	Fuzzy uses partial membership therefore more useful for real problems.	Determining membership function is no easy.
(v)	Watershed method	This method is based on topological interpretation.	Results are more stable, detected boundaries are continuous.	Complex calculation of gradients.
(vi)	PDF based method	This method is based on the working of differential equations.	Fastest method, best for time critical applications.	More computational complexity.
(vii)	ANN based method	This method is based on the simulation of learning process for decision making.	No need to write complex programs.	More wastage of tim in training.

Q.26. Discuss about the connected components.

Ans. A set of pixels in which each pixel is connected to all other pixels i called a connected component.

The set of all connected components of $\overline{A}$ (the complement of A) that hav points on the border of an image is called the background. All other components of $\overline{A}$ are called holes. Let us consider the simple picture shown below –

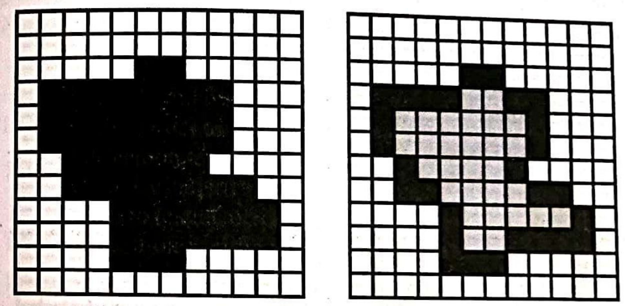
	1	
1		1
	1	

If we consider 4-connectedness for both foreground and background, there are four objects that are 1 pixel in size and there is one hole. If we use 8-connectedness, then there is one object and no hole. Intuitively, in both cases we have an ambiguous situation. A similar ambiguous situation arises in a simple case like –

1	0
0	1

where if the 1s are connected, then the 0s should not be.

To avoid this situation, different connectedness should be used for objects and background. If we use 8-connectedness for A, then 4-connectedness should be used for $\overline{A}$. An example of a simple binary image with its boundary, interior and surrounding is shown in fig. 2.20.



(a) Original Image *(b)* *Boundary Pixels* *Interior Pixels* *Surrounding Pixels*

Fig. 2.20 A Binary Image with its Boundary, Interior, and Surroundings

The *boundary* of A is the set of pixels of A that have 4-neighbours in $\overline{A}$. The boundary is usually denoted by A'. The *interior* is the set of pixels of A that are not in its boundary. The interior of A is $(A - A')$. Region T *surrounds* region A (or A is inside T), if any 4-path from any point of A to the border of the picture must intersect T.

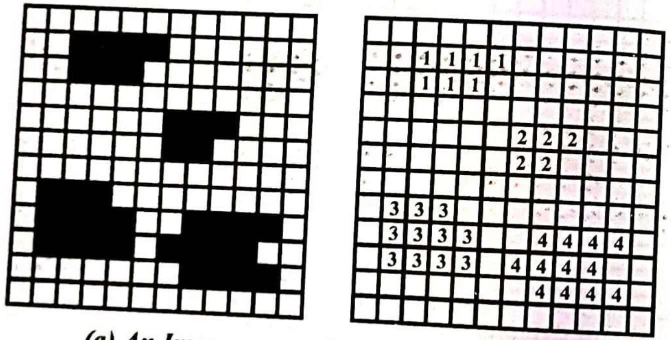
Q.27. Explain the term connected components labeling.

Ans. "A connected components labeling of a binary image B is a labeled image LI in which the value of each pixel is the label of its connected component".

One of the most common operations in machine vision is finding the connected components in an image. The points in a connected component form a candidate region for representing an object. As mentioned earlier, in computer

vision most objects have surfaces. Points belonging to a surface project spatially close points. The notion of "spatially close" is captured by connected components in digital images. However, the connected component algorithms usually form a bottleneck in a binary vision system. The algorithm is sequential in nature, because the operation of finding connected components is a global operation. If there is only one object in an image, then there may not be a need for finding the connected compoents; however, if there are many objects in an image and the object properties and locations need to be found, then the connected components must be determined.

A component labeling algorithm finds all connected components in an image and assigns a unique label to all points in the same component. Fig. 2.21 (a) and (b) shows an image and its labeled connected components. In many applications, it is desirable to compute characteristics such as size, position, orientation, and bounding rectangle of the components while labeling the components.



(a) An Image (b) Connected Component Image

Fig. 2.21

Q.28. Explain the recursive connected component algorithm.

Ans. There are a number of different algorithms for the connected component labeling operation. Some algorithms assume that the entire image can fit in memory and employ a simple, recursive algorithm that works on one component at a time, but can move all over the image while doing so.

Suppose that B is a binary image with MaxRow + 1 rows and MaxCol + 1 columns. We wish to find the connected components of the 1-pixels and produce a labeled output image LI in which every pixel is assigned the label of its connected component. The strategy, adapted from the Tanimoto AI text, is to first negate the binary image, so that all the 1-pixels become – 1's. This is needed to distinguish unprocessed pixels (– 1) from those of component label 1. This can be done using a function called *negate* that inputs the binary image B and outputs the negated image LI, which will become the labeled image. Then the process of finding the connected

components becomes one of finding a pixel whose value is – 1 in LI, assigning it a new label, and calling procedure *search* to find its neighbours that have value – 1 and recursively repeat the process for these neighbours. The utility function neighbours (L, N) is given a pixel position defined by L and N. It returns the set of pixel positions of all of its neighbours, using either the 4-neighbourhood or 8-neighbourhood definition. Only neighbours that represent legal positions on the binary image are returned. The neighbours are returned in scan-line order as shown in fig. 2.22.

	1	
2	*	3
	4	

(a) Four-neighbourhood

1	2	3
4	*	5
6	7	8

(b) Eight-neighbourhood

Fig. 2.22 Scan-line Order for Returning the Neighbours of a Pixel

Algorithm for Recursive Connected Components – Compute the connected components of a binary image. Here, B is the original binary image and LI will be the labeled connected component image.

```
procedure recursive_connected_components (B, LI);
{
LI := negate(B);
label := 0;
find_components(LI, label);
print(LI);
}
procedure find_components(LI, label);
{
for L := 0 to MaxRow
    for N := 0 to MaxCol
        if LI[L, N] = = – 1 then
            {
            label := label + 1;
            search(LI, label, L, N);
            }
}
procedure search(LI, label, L, N);
```

```
{
LI[L, N] := label;
Nset :=neighbours(L, N);
for each (L', N') in Nset
    {
    if LI[L', N'] = = – 1
    then search(LI, label, L', N');
    }
}
```

The first five steps of the recursive labeling algorithm applied to the first component of the binary image of fig. 2.23. The image shown is the (partially) labeled image LI. The boldface pixel of the image is the one being processed by the search procedure. Using the neighbourhood orderings shown in fig. 2.22, the first unprocessed neighbour of the boldface pixel whose value is –1 is selected at each step as the next pixel to be processed.

Step 1.

–1	–1	0	–1	–1	–1
–1	–1	0	–1	0	0
–1	–1	–1	–1	0	0

Step 2.

1	–1	0	–1	–1	–1
–1	–1	0	–1	0	0
–1	–1	–1	–1	0	0

Step 3.

1	1	0	–1	–1	–1
–1	–1	0	–1	0	0
–1	–1	–1	–1	0	0

Step 4.

1	1	0	–1	–1	–1
1	–1	0	–1	0	0
–1	–1	–1	–1	0	0

Step 5.

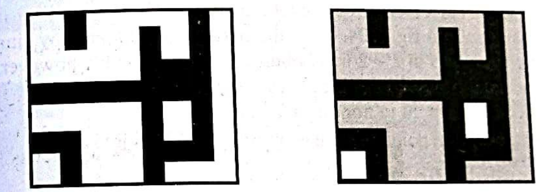
1	1	0	–1	–1	–1
1	1	0	–1	0	0
–1	–1	–1	–1	0	0

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

(a) Binary Image

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
3	3	3	3	0	4	0	2
0	0	0	3	0	4	0	2
5	5	0	3	0	0	0	2
5	5	0	3	0	2	2	2

(b) Connected Components Labeling



(c) Binary Image and Labeling, Expanded for Viewing

Fig. 2.23 A Binary Image with Five Connected Components of the Value 1

Q.29. Write short note on row-by-row labeling algorithm.

Ans. The classical algorithm, deemed so because it is based on the classical connected components algorithm for graphs, was described in Rosenfeld and Pfaltz in 1966. The algorithm makes two passes over the image – one pass to record equivalences and assign temporary labels and the second to replace each temporary label by the label of its equivalence class. In between the two passes, the recorded set of equivalence, stored as a binary relation, is processed to determine the equivalence classes of the relation. Since that time, the union-find algorithm, which dynamically constructs the equivalence classes as the equivalences are found, has been widely used in computer science applications. The union-find data structure allows efficient construction and manipulation of equivalence classes represented by tree structures. The addition of this data structure is a useful improvement to the classical algorithm.

Q.30. Discuss about the hierarchical image representation.

Ans. In segmentation an image can be represented at many different resolutions. By reducing an image's resolution, i.e. reducing the size of the array, some data is lost, making it more difficult to recover information. However, reduction in resolution results in reduced memory and computing requirements. Hierarchical representation of images allows representation at multiple resolutions. In many applications, one can compute properties of

images first at a low resolution and then perform additional computations over a selected area of the image at a higher resolution. Hierarchical representations are also used for browsing in images. Two commonly used forms of hierarchical representations are pyramids and quad trees.

(i) Pyramids – A pyramid representation of an $n \times n$ image contains the image and k reduced versions of the image. Usually n is a power of 2 and the other images are $n/2 \times n/2$, $n/4 \times n/4$, ... 1×1. In a pyramid representation of an image, the pixel at level l is obtained by combining information from several pixels in the image at level $l + 1$. The whole image is represented as a single pixel at the top level, level 0, and the bottom level is the original (unreduced) image. A pixel at a level represents aggregate information represented by several pixels at the next level. An image and its reduced versions in a pyramid is shown in fig. 2.24. Here the pyramid is obtained by simply averaging the gray values in 2×2 neighbourhoods. It is possible however, to devise other strategies to form reduced-resolution versions. Similarly, it is possible to taper the pyramid in nonlinear ways.

An implementational point is that the entire pyramid fits into a linear array of size $2(2^{2 \times level})$.

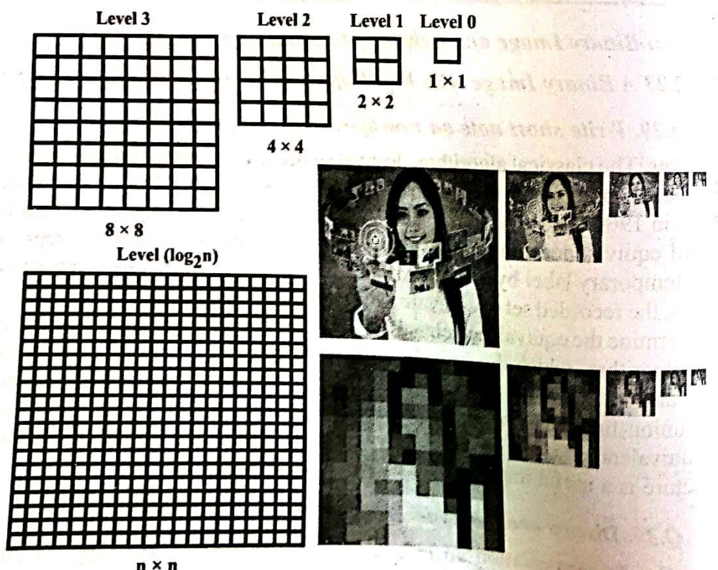
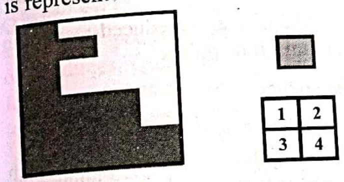


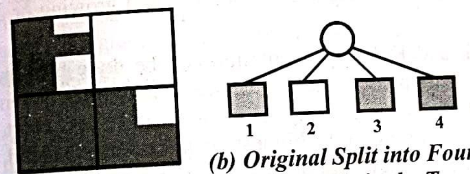
Fig. 2.24

(ii) Quad Trees – A ***quad tree*** may be considered as an extension of pyramids for binary images. A quad tree contains three types of nodes – white, black and gray. A quad tree is obtained by recursive splitting of an image.

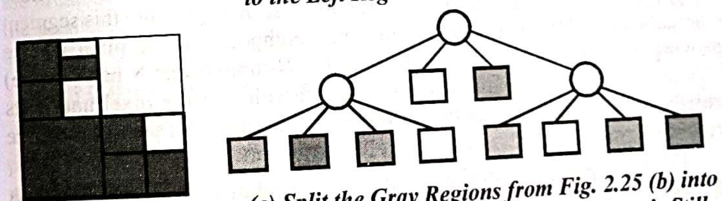
A region in an image is split into four subregions of identical size, as shown in fig. 2.25. For each subregion, if all points in the region are either white or black, then this region is no longer considered as a candidate for splitting; if it contains pixels of both kinds, it is considered to be a "gray region" and is further split into four subregions. An image obtained using this recursive splitting is represented in a tree structure. The splitting process is repeated until there



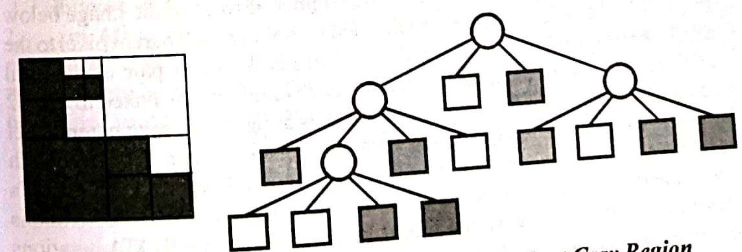
(a) Original Image, "Gray Region"



(b) Original Split into Four Subregions (The Left Node in the Tree Corresponds to the Left Region in the Image)



(c) Split the Gray Regions from Fig. 2.25 (b) into Four Subregions. One of these Regions is Still a Gray Region



(d) Splitting of the Last Gray Region and the Final Quad Tree

Fig. 2.25 The Building of a Quad Tree

are no gray regions in the tree. Each node in this structure is either a leaf node or has four children – thus the name quad tree.

Quad trees are finding increasing application in spatial databases. Several algorithms have been developed for converting a raster array to a quad tree and a quad tree to a raster array. Algorithms for computing several pictorial properties have also been developed.

Fig. 2.24 the original image is a 512×512 image, its reduced-resolution versions are successively obtained by averaging four points.

Q.31. Write short note on spatial clustering.

Ans. It is possible to determine the image segments by simultaneously combining clustering in measurement space with spatial region growing. Such a technique is called spatial clustering. In essence, spatial clustering schemes combine the histogram mode seeking technique with a region growing or a spatial linkage technique.

According to Haralick and Kelly, segmentation can be done by first locating, in turn, all the peaks in the measurement space histogram, and then determining all pixel locations having a measurement on the peak. Next, beginning with a pixel corresponding to the highest peak not yet processed, both spatial and measurement space region growing are simultaneously performed in the following manner. Initially, each segment is the pixel whose value is on the current peak. Consider for possible inclusion into this segment a neighbour of this pixel (in general, the neighbours of the pixel we are growing from) if the value of a neighbour (an N-tuple for an N band image) is close enough in measurement space to the value of the pixel and if its probability is not larger than the probability of the value of the pixel we are growing from.

Another spatial clustering scheme is a spatial pyramid constrained ISODATA kind of clustering. The bottom layer of the pyramid is the original image. Each successively higher layer of the pyramid is an image having half the number of pixels per row and half the number of rows of the image below it. Initial links between layers are established by linking each parent pixel to the spatially corresponding 4×4 block of child pixels. Each pair of adjacent parent pixels has 8 child pixels in common. Each child pixel is linked to a 2×2 block of parent pixels. The iterations proceed by assigning to each parent pixel the average of its child pixels. Then each child pixel compares its value with each of its parent's values and links itself to its closest parent. Each parent's new value is the average of the children to which it is linked, etc. The iterations converge reasonably quickly and for the same reason the ISODATA iterations converge. If the top layer of the pyramid is a 2×2 block of great-grandparents, then there are at most 4 segments which are the respective great-grandchildren of these 4 great-grandparents.

Q.32. Explain the method of region splitting and merging for region segmentation.

Ans. Assume that, R is the whole image region and choose a predicate Q. One method for segmenting R is to subdivide it successively into smaller and smaller quadrant regions so that, for any region R_i, $Q(R_i)$ = TRUE. We begin with the whole region. If Q(R) = FALSE, the image is divided into quadrants. If Q is FALSE for any quadrant, the quadrant is subdivided into subquadrants, and so on. This specific splitting method has a convenient representation in the form of quadtrees. Quadtrees represent trees in which each node includes exactly four descendants as shown in fig. 2.19. Note that the root of the tree corresponds to the whole image, and that each node corresponds to the subdivision of a node into four descendant nodes. In this situation, only R_4 was subdivided further. If only splitting is employed, the final partition includes adjacent regions with identical properties. By permitting merging as well as splitting, this disadvantage can be remedied. Satisfying the constraints of segmentation requires merging only adjacent regions whose combined pixels satisfy the predicate Q. That is, two adjacent regions R_j and R_k are merged only if $Q(R_j \cup R_k)$ = TRUE.

It is customary to satisfy a minimum quadregion size beyond which no further splitting is carried out. Numerous variations of the preceding fundamental theme are possible. For example, a significant simplification results if we permit merging of any two adjacent regions R_i and R_j if each one satisfies the predicate individually. This results in a much simpler way due to testing of the predicate is limited to individual quadregions. This simplification is still capable of yielding good segmentation results.

Q.33. Write down the split and merge algorithm for region segmentation.

Ans. Split and merge operations may be used together. After a presegmentation based on thresholding, a succession of splits and merges may be applied to refine the segmentation. Combined split and merge algorithms are useful for segmenting complex scenes. Domain knowledge may be introduced to guide the split and merge operations.

Suppose that an image is partitioned into a set of regions, $\{R_n\}$, for n = 1, 2, ..., m. All of the pixels in a region will be homogeneous according to some property defined by a predicate A applied to the region. The predicate represents the similarity between the pixels in a region. For example, the predicate could be defined using the variance in gray values within a region –

$$A(R) = \begin{cases} 1 & \text{if the variance is small} \\ 0 & \text{otherwise} \end{cases}$$

Algorithm for Split and Merge Region Segmentation –

Step 1 – Start with the entire image as a single region.

Step 2 – Pick a region R. If A(R) is false, then split the region into four regions.

Scanned with CamScanner

Step 3 – Consider any two or more neighbouring subregions, $R_1, R_2, \ldots, R_n$ in the image. If $A(R_1 \cup R_2 \cup \ldots \cup R_n)$ is true, merge the n regions into a single region.

Step 4 – Repeat these steps until no further splits or merges take place.

Q.34. Discuss about the rule based segmentation in image segmentation.

Ans. A rule based expert system for segmentation introduced by Nazif and Levine in 1984. According to this approach, system contains a set of processes such as the line analyzer, the initializer, the region analyzer, the area analyzer, the focus of attention, the scheduler, and two associate memories, i.e., the long term memories (LTM) and short term memories (STM). The short term memories holds the input image, the segmentation data and the output. Other hand, LTM contains the model representing the system knowledge about low-level segmentation and control strategies. A system process matches rules in the LTM against the data stored in the STM. When a match occurs, the rule fires, and an action, usually involving data modification is performed. According to this approach, the model stored in the LTM has three levels of rules – knowledge rules, the control rule and the highest rules.

(i) Knowledge Rules – Encode information about the properties of regions, lines, and area in the form of situation-action pairs. The specific actions include splitting a region, merging two regions, adding, deleting, or extending a line, merging two lines, and modifying focus of attention area. These rules are classified by their action.

(ii) Control Rules – These rules are divided into two categories – focus-of-attention and inference rules. Focus-of-attention rules find the next data entry to be considered, a region, a line, or an entire area. These rules control the focus-of-attention strategy. The inference rules are metarules in that their actions don't modify the data in the short term memory. Instead, they alter the matching order of different knowledge rule sets.

(iii) Highest Rule Level – These rules are strategy rules that select the set of control rules that executes the most appropriate control strategy for a given set of data.

The conditions of the rules base are made up of first, a symbolic qualifier depicting a logical operation to be performed on the data, second, a symbol denoting the data entry on which the condition is to be matched. Third, a feature of this data entry, fourth, an optional NOT qualifier, and last fifth, an optional DIFFERENCE qualifier that applies the operation to differences in feature values. Rule based segmentation is useful because it is general but allow more specific strategies to be incorporated without changing the code.

Q.35. Describe the motion based segmentation.

Ans. Motion is a very powerful cue used by humans and many other animals to extract objects or regions of interest from a background of irrelevant detail. In imaging application motion arises from a relative displacement between the sensing system and the scene being viewed such as in robotic applications autonomous navigation and dynamic scene analysis. The use of motion in segmentation, we used spatial techniques. One of the simplest approaches for detecting changes between two image frames $f(a, b, c_i)$ and $f(a, b\ c_j)$ taken at times c_i and c_j respectively, is to compare the two images pixel by pixel. One procedure for doing this is to form a difference image. Assume that we have a reference image containing only stationary components. Comparing this image against a subsequent image of the same scene, but including a moving object results in the difference of the two images canceling the stationary elements, leaving only nonzero entries that correspond to the non stationary image components.

A difference image between two images taken at times c_i and c_j may be defined as

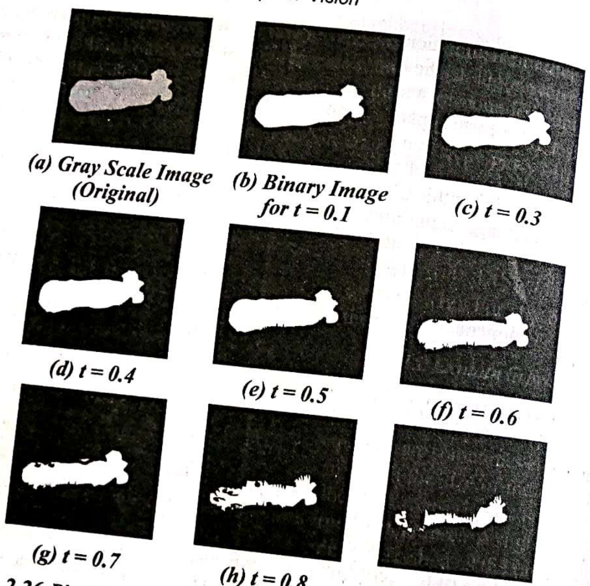
$$D_{ij}(a, b) = \begin{cases} 1, & \text{if } |f(a,b,c_i) - f(a,b,c_j)| > T \\ 0, & \text{otherwise} \end{cases} \quad \ldots(i)$$

Here, T is a specified threshold. $D_{ij}(a, b)$ has a value of 1 at spatial coordinates (a, b) only if the intensity difference between two images is appreciably different at those coordinates, as determined by the specified threshold T. It is assumed that all images are of the same size finally we note that the values of the coordinates (a, b) in equation (i) span the dimensions of these images, so that the difference image $D_{ij}(a, b)$ is of the same size as the images in the sequence.

AREA EXTRACTION – CONCEPTS, DATA STRUCTURES, EDGE LINE LINKING, HOUGH TRANSFORM, LINE FITTING, CURVE FITTING (LEAST-SQUARE FITTING)

Q.36. How do threshold values affect conventional area extraction ?

Ans. In image processing, the conventional method for extracting area is to convert the gray scale image to binary image by setting a threshold value. With pixel values greater than the threshold value separated from those lower than it, the result would be an image showing a white blob corresponding to the shape of the animal on a black background. However, the selection of the threshold value is critical in determining the area of the pig area. If the threshold value is too high, some darker parts of the pig will be missing, so the calculated pig area will be smaller than the actual area. If the threshold value is too low, some brighter parts of the environment could be counted as part of the pig, causing the calculated pig area to be larger than the actual area. Fig. 2.26 shows the binary images of a pig corresponding to different threshold values.



(a) Gray Scale Image (Original) (b) Binary Image for $t = 0.1$ (c) $t = 0.3$

(d) $t = 0.4$ (e) $t = 0.5$ (f) $t = 0.6$

(g) $t = 0.7$ (h) $t = 0.8$ (i) $t = 0.9$

Fig. 2.26 Pig Binary Image as a Function of Threshold Value t

Pig areas at different threshold values were calculated and plotted in fig. 2.27. It shows that the area decreased monotonically with the increase of the threshold value. In order to obtain a relatively accurate extraction of the area, threshold values were often determined manually through human observation. However, the limit of human eye's resolution will possibly bring in errors during the measurement.

In fig. 2.27, pig area is extracted by converting gray scale image to binary image as a function of the threshold value. The arrow in the figure indicates the corresponding threshold value in the edge detection method.

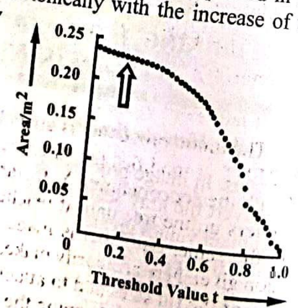


Fig. 2.27

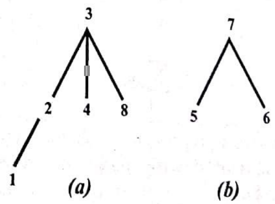
Q.37. What do you mean by union-find structure ?

Ans. The union-find data structure is to store a collection of disjoint sets and to efficiently implement the operations of union (merging two sets into one) and find (determining which set a particular element is in). Each set is stored as a tree structure in which a node of the tree represents a label and points to its one parent node. This is accomplished with only a vector array PARENT whose subscripts are the set of possible labels and whose values are the labels of the parent nodes. A parent value of zero means that this node is the root of the tree. For example, the union-find data structure for two sets of labels is shown in fig. 2.28. The first set contains the labels {1, 2, 3, 4, 8}, and the second set contains labels {5, 6, 7}. For each integer label i, the value of PARENT (i) is the label of the parent of i or zero if i is a root node and has no parent. Label 3 is the parent node and set label for the first set; label 7 is the parent node and set label for the second set. The values in array PARENT tell us that nodes 3 and 7 have no parents, label 2 is the parent of label 1, label 3 is the parent of labels 2, 4 and 8, and so on. Note that element 0 of the array is not used, since 0 represents the background label, and a value of 0 in the array means that a node has no parent.

The find procedure is given a label X and the parent array PARENT. It merely follows the parent pointers up the tree to find the label of the root node of the tree that X is in. The algorithm of find as follows –

Find the parent label of a set. Here, X is a label of the set. PARENT is the array containing the union-find data structure.

```
procedure find(X, PARENT);
{
j := X;
while PARENT[j] <> 0
    j := PARENT[j];
return(j);
}
```



PARENT

1	2	3	4	5	6	7	8
2	3	0	3	7	7	0	3

Fig. 2.28

The union procedure is given for two labels X and Y and the parent array PARENT. It modifies the structure (if necessary) to merge the set containing X with the set containing Y. It starts at labels X and Y and follows the parent pointers up the tree until it reaches the roots of the two sets. If the roots are not the same, one label is made the parent of the other. The procedure for union given here arbitrarily makes X the parent of Y. It is also possible to keep track of the set sizes and to attach the smaller set to the root of the larger set; this has the effect of keeping the tree depths down. The algorithm of union as follows –

Construct the union of two sets. Here, X is the label of the first set. Y is the label of the second set and PARENT is the array containing the union-find data structure.

```
procedure union(X, Y, PARENT);
{
j := X;
k := Y;
while PARENT[j] <> 0
    j := PARENT[j];
while PARENT[k] <> 0
    k := PARENT[k];
if j <> k then PARENT[k] := j;
}
```

Q.38. How to detect discontinuities by point detection, line detection and edge detection method ? ***(R.G.P.V., June 2017)***

Ans. There are several methods to detect the three basic types of gray level discontinuities. The first one is point detection, the second one is line detection and the third one is edge detection. Running a mask through the image is the most common method to look for discontinuities. In this method, the sum of products of the coefficients with the gray levels contained in the region encompassed by the mask are calculated. The response of a 3 × 3 mask at any point in the image is given by the following equations –

$$R = w_1z_1 + w_2z_2 + w_3z_3 + ... + w_9z_9$$
$$= \sum_{i=1}^{9} w_i z_i \qquad ...(i)$$

where w_i represents mask coefficient and z_i represents gray level of the pixel associated with mask coefficient. The mask response is specified in terms of center location.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Fig. 2.29 Representation of General 3 × 3 Mask

(i) Point Detection – In principle, isolated points detection in an image is very simple. At a location on which the mask is centered a point is detected, if

$$|R| \geq T$$

where T represents a non negative threshold.

This equation is used to calculate the weighted differences between the center point and its neighbours. The concept of this method is that an isolated point will be different from its surroundings and hence can be easily detected by this type of mask. If the sum of mask coefficients is zero, then the mask response will be zero in areas of constant gray level.

–1	–1	–1
–1	8	–1
–1	–1	–1

Fig. 2.30 Representation of Point Detection Mask

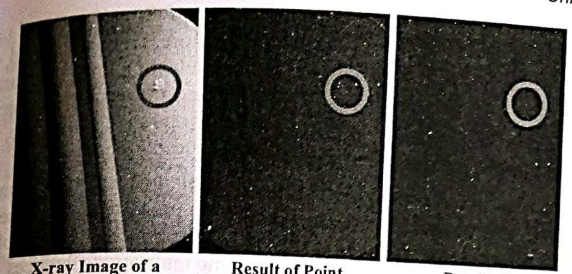


Fig. 2.31 Representation of Point Detection

(ii) Line Detection – Line detection is more complex then point detection. Considering the masks represented in fig. 2.32. When the first mask were moved around an image, this mask would respond very strongly to lines oriented horizontally. When the line passed through the middle row of the mask the maximum response would be obtained in case of a constant background. By drawing a simple array of 1's with a line of different gray level running horizontally through the array, this can be verified. The second mask responds best to lines oriented at +45° and third mask to vertical lines and the fourth mask to lines in the – 45° direction. These directions may be established also by noting that the preferred direction of each mask is weighted with a larger coefficient as compared to other possible directions. Let R_1, R_2, R_3, R_4 represent the masks' responses in fig. 2.32, from left to right, and the

(a) Lines Oriented Horizontally

–3	–3	–3
2	2	2
–3	–3	–3

(b) Lines Oriented at + 45°

–3	–3	2
–3	2	–3
2	–3	–3

(c) Lines Oriented Vertically

–3	2	–3
–3	2	–3
–3	2	–3

(d) Lines Oriented at – 45°

2	–3	–3
–3	2	–3
–3	–3	2

Fig. 2.32 Representation of Line Detection Mask

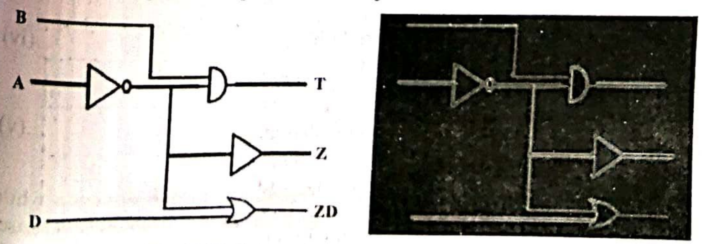


Fig. 2.33 Representation of Line Detection

four masks are run individually through an image. If $|R_i| > |R_j|$, for all $j \neq i$, at certain point in the image the point is said to be more likely associated with a line in the direction of mask i. Alternatively, for detecting lines in a particular direction, the mask attached with that direction is used and its output is threshold. The remaining points are strongest responses for one pixel thick line. These points correspond nearest to the direction specified by the mask.

(iii) Edge Detection – The process of obtaining meaningful transitions in an image is known as edge detection. In image processing, it is one of the central tasks of the lower level. Points where brightness changes sharply form the border between different objects. Determining intensity differences in local image regions such points can be detected. It means that the edge-detection algorithm should look for a neighbourhood with strong signs of change. Various edge detectors work based on calculating the intensity gradient at a point in the image. Edge detection is used to recognize areas of an image where a major change in intensity encounters. These changes are mostly related with some physical boundary in the scene from which the image is derived. Edges characterise object boundaries typically in images. Edges are used for identification, segmentation and registration of objects in a scene.

The methods of edge detection are as follows –

(a) Prewitt Mask – It uses the basic concept of central difference. Assume the pixels arrangement with respect to the central pixel [i, j] as follows –

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & [i, j] & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix} \quad ...(i)$$

The Prewitt edge detector partial derivatives are computed as given below-

$$G_X = (a_2 + Ca_3 + a_4) - (a_0 + Ca_7 + a_6) \quad ...(ii)$$

and

$$G_Y = (a_6 + Ca_5 + a_4) - (a_0 + Ca_1 + a_2) \quad ...(iii)$$

The constant C represents the emphasis provided to pixels nearer to the centre of mask.

G_X and G_Y represents the approximations at [i, j].

Putting C = 1, the Prewitt edge detector mask becomes

$$G_X = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad ...(iv)$$

$$G_Y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad ...(v)$$

The Prewitt edge detector mask differentiates in one direction while averages in other direction. Hence, edge detector is less vulnerable to noise. The Prewitt edge detector masks have longer support.

(b) Sobel Mask – It was developed by Irwin Sobel. It uses the basic concept of central differences but gives greater weight to the central pixels during averaging. The Sobel mask can be considered as 3 × 3 approximations to first derivatives of Gaussian kernels.

For the pixels arrangement shown in equation (i), Sobel edge detector partial derivatives are computed as given below –

$$G_X = (a_2 + 2a_3 + a_4) - (a_0 + 2a_7 + a_6) \quad ...(vi)$$

and

$$G_Y = (a_6 + 2a_5 + a_4) - (a_0 + 2a_1 + a_2) \quad ...(vii)$$

The Sobel edge detector mask is as follows –

$$G_X = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad ...(viii)$$

and

$$G_Y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad ...(ix)$$

The noise suppression feature of a Sobel mask is better as compared to the Prewitt mask.

Q.39. Explain the difference between edge and line with graph.
(R.G.P.V., May 2019)

Ans. Refer to Q.38.

Edges, lines, and points carry a lot of information about the various regions in the image. These features are usually termed as local features, since they are extracted from the local property alone. Though the edges and lines are both detected from the abrupt change in the gray level, yet there is an important difference between the two. An edge essentially demarcates between two distinctly different regions, which means that an edge is the border between two different regions. A line, on the other hand, may be embedded inside a single uniformly homogeneous region. For example, a thin line may run between two plots of agricultural land, bearing the same vegetation. A point is embedded

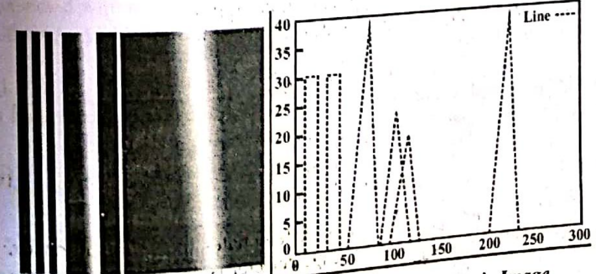


Fig. 2.34 Edge Profile of One Row of a Synthetic Image

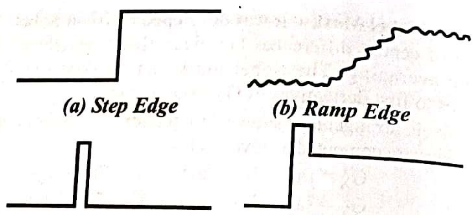


Fig. 2.35 Different Types of Edges

inside a uniformly homogeneous region and its gray value is different from the average gray value of the region in which it is embedded. This is analogous to a spike. The changes in the gray levels in case of a perfect step edge, line, ramp edge are shown in the form of an edge profile in fig. 2.34. The diverse forms and nature of ideal edges and lines, such as step edge, ramp edge, line, step line, are shown in fig. 2.35.

Q.40. Explain local analysis method of edge linking and boundary detection.

Ans. Ideally, pixels lie only on edges but in actual practice, set of pixels seldom characterizes an edge completely due to noise, breaks in the edge from nonuniform illumination and other effects which result in spurious intensity discontinuities. Hence, to assemble edge pixels into meaningful edges, edge detection algorithms are followed by edge linking methods.

Local Processing Method of Edge Linking – Analyze the characteristics of pixels in a small neighbourhood about every point (x, y) in an image is the simple method for linking edge points. According to a set of predefined criteria, all similar points are linked, developing an edge. There are following two properties for establishing similarity of edge pixels in this analysis –

(i) The strength of the gradient operator response is used to generate the edge pixel.

(ii) Gradient vector direction.

The value of ∇x provides first property as given below –

$$\nabla x \approx |G_X| + |G_Y| \quad ...(i)$$

An edge pixel with coordinates (x_0, y_0) in a predefined neighbourhood of (x, y), is similar in magnitude to the pixel at (x, y), if

$$|\nabla x(x, y) - \nabla x(x_0, y_0)| \le E \quad ...(ii)$$

where E represents non-negative threshold.

Following expression represents the direction of the gradient vector –

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_Y}{G_X}\right) \quad ...(iii)$$

An edge pixel at (x_0, y_0) in the predefined neighbourhood of (x, y) has an angle same as to the pixel at (x, y), if

$$|\alpha(x, y) - \alpha(x_0, y_0)| < A \quad ...(iv)$$

where A is a non-negative angle threshold.

From equation (iii), we can see the direction of the edge is perpendicular to the gradient vector direction at (x, y). If direction and magnitude criteria are satisfied, a point in the predefined neighbourhood of coordinates (x, y) is linked to the pixel at that coordinates. At every position in the image, this procedure is repeated. To maintain a record of the linked points as the center of the neighbourhood is shifted from one pixel to another pixel assign a different gray level to each set of linked edge pixels.

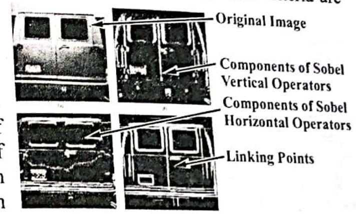


Fig. 2.36 Representation of Edge Linking

Q.41. Discuss global processing via Hough transforms and graph theoretic techniques. ***(R.G.P.V., June 2017)***

Ans. **Global Processing via Hough Transform** – This method was developed by Hough in 1962. At a point (x_i, y_i) the general equation of a straight line in slope-intercept form is given below –

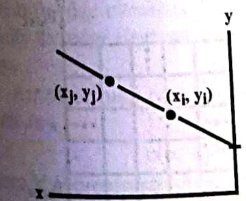
$$y_i = mx_i + c \quad ...(i)$$

Generally, infinite lines may pass via point $(x_i\ y_i)$. For varying values of m and c, they all satisfy the equation (i). Equation (i) can also be written as

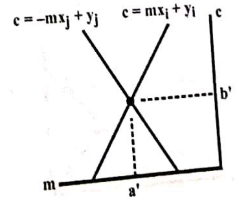
$$c = y_i - mx_i$$

$$c = -mx_i + y_i \quad ...(ii)$$

Assume that mc-plane provides equation of a single line for a fixed pair (x_i, y_i). In addition, a second point (x_j, y_j) also has a line in parameter space connected with it. And at a point (a', b') this line intersects the line associated with (x_i, y_i). Where a' represents the slope and b' represents intercept of the



(a) xy-plane Representation



(b) Parameter Space Representation

Fig. 2.37

line containing both (x_i, y_i) and (x_j, y_j) points in the xy-plane. In fact, all points contained on this line have lines in parameter space which intersect at (a', b'). These concepts are shown in fig. 2.37.

Hough transform is more popular because is allows subdividing the parameter space into accumulator cells shown in fig. 2.38, where c_{max} and c_{min} represent expected range of intercept values and m_{max} and m_{min} represent expected range of slope. The cell at coordinates (i, j) with accumulator value A(i, j) corresponds to the square which is related with parameter space coordinates (m_i, c_j). At the start, these cells are set to zero value. Then, for every point (x_k, y_k), let the parameter m equal to each of the permitted subdivision values on the m-axis and corresponding c value can be obtained from equation $c = -mx_k + y_k$. Then output values of c are rounded off to the closest permitted value in the c-axis. When a selection of m_p outputs in solution c_q, we let A(p, q) = A(p, q) + 1. At the end, a value of Q in A(i, j) corresponds to Q points in the xy-plane lying on the line $y = m_i x + c_j$. The accuracy of the colinearity of these points is determined by the number of subdivisions in the mc-plane. For every point (x_k, y_k), k value of c corresponding to the k possible values of m can be obtained by subdividing the m-axis into k increments. This process requires nk computations with n image points. Hence this method is linear in n and the multiplication nk does not approach the number of computations unless k approaches or exceeds n. The problem with using the equation y = mx + c is to show a line for which the slope approaches infinity as the line approaches the vertical. To solve this problem the normal representation of line is given as below –

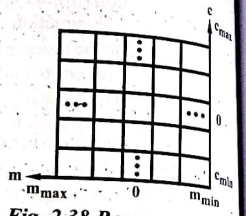
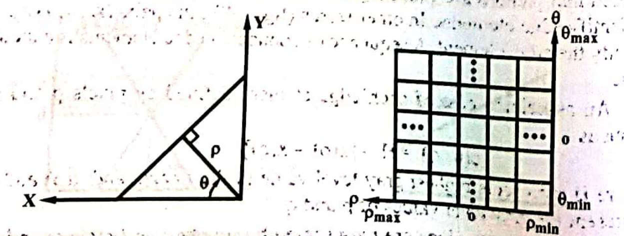


Fig. 2.38 Representation of Accumulator Cells

$$x \cos\theta + y \sin\theta = \rho \qquad \text{...(iii)}$$

In constructing accumulator table, the use of this representation is identical to the method discussed for the slope-intercept representation. However, the



(a) Normal Line Representation *(b) ρθ-plane Subdivided into Cells*

Fig. 2.39

loci are sinusoidal curves in place of straight lines in the ρθ-plane. As before Q collinear points lying on a line $x \cos\theta_j + y \sin\theta_j = \rho_i$ yield Q sinusoidal curves which intersect at (ρ_i, θ_j) in the parameter space. For increasing values of θ and obtaining corresponding ρ gives Q entries in the accumulator A(i, j) associated with the cell (ρ_i, θ_j). The limit of angle θ is + 90° to –90° measured with respect to the x-axis. Thus from fig. 2.39 (a), a horizontal line will have θ = 0°, with ρ being equal to the positive x-intercept. In a similar way, a vertical line will have θ = 90°, with ρ being equal to the positive y-intercept or a vertical line includes θ = –90°, with ρ being equal to the negative y-intercept as [shown in fig. 2.39 (a)] according to equation (iii).

Global Processing via Graph-Theoretic Methods – It is a global approach for edge detection and linking based on showing edge segments in the form of a graph and finding the graph for low cost paths corresponding to significant edges. This representation gives a rugged method. Which works well in presence of noise. However, the method is more difficult and needs more processing time. A finite, non empty set of nodes N with a set U of unordered pairs of distinct element of N is known as graph G = (N, U). Arc refers to a each pair (n_i, n_j) of U. A directed graph refer as a graph G in which the arcs are directed. The n_j is known as a successor of the parent node n_i if an arc is directed from node n_i to n_j. Node expansion refers to the process of identifying the successors of a node. Level is specified in each graph. Level zero has a single node which is known as the root node. The nodes in last level are known as goal nodes. With each arc (n_i, n_j), a cost $c(n_i, n_j)$ may be attached. A sequence of nodes n_1 to n_k with each node n_i being a successor of node n_{i-1} is known as a path from n_0 to n_k. Following expression represents the cost of the whole path –

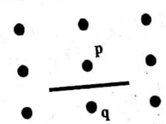


Fig. 2.40 Representation of Edge Element between Pixels p and q

$$c = \sum_{i=2}^{k} c(n_{i-1}, n_i) \qquad \text{...(i)}$$

Let us define an edge element as the boundary between two pixels p and q, such that p and q are four neighbours. The xy-coordinates of points p and q identify edge elements. In other words (fig. 2.40), the pairs (x_p, y_p) (x_q, y_q) specify the edge element. A sequence of connected edge elements is called an edge.

An associated cost of each edge element defined by pixels p and q is given as –

$$c(p, q) = H - [x(p) - x(q)] \qquad \text{...(ii)}$$

where H represents highest gray level value in the image, and x(p) and x(q) represent the gray level values of p and q.

The gray level values are represented by numbers in brackets and pixel coordinates are represented by the numbers which are outside the brackets.

Generally, problem of obtaining a minimum cost path is not trivial. Typically, the method has to sacrifice optimally for the sake of speed. Following algorithm shows a class of method which use heuristics in manner to decrease the effort of finding. Let, r(n) represents an estimate of the cost of a minimum cost path from the start node o to a goal node, where the path is constrained to go via n. This cost may be defined as an estimate of the cost of a minimum cost path from 0 to n plus an estimate of cost of that path from n to a goal node i.e.

$$r(n) = g(n) + h(n) \quad \text{...(iii)}$$

Here g(n) represents the lowest cost path from o to n found so far, and using any available heuristic information we can obtain h(n). Following algorithm uses r(n) as the basis for performing a graph search –

(i) Start

(ii) Indicate the start node **open** and set g(o) = 0.

(iii) Exit with failure if no node is **open else** continue.

(iv) Indicate **closed** the **open** node n **IF** estimate r(n) of open node n is smallest.

(v) Exit with the solution path achieved by tracing back through the pointers **IF** n is a goal node **ELSE** continue.

(vi) Expand node n, producing all of its successors. **IF** there are no successors then **(GoTo)** step (iii).

(vii) **IF** a successor n_i is not indicated, set

$$r(n_i) = g(n) + c(n, n_i)$$

and indicate it **open**, and direct pointers from it back to n.

(viii) **IF** a successor n_i is indicated **closed** or **open**, update its value by–

$$g'(n_i) = \min[g(n_i), g(n) + c(n, n_i)]$$

Indicate **open** those **closed** successors whose g' values were thus lowered and redirect to n the pointers from all nodes whose g' values were lowered and GoTo step (iii).

(ix) End

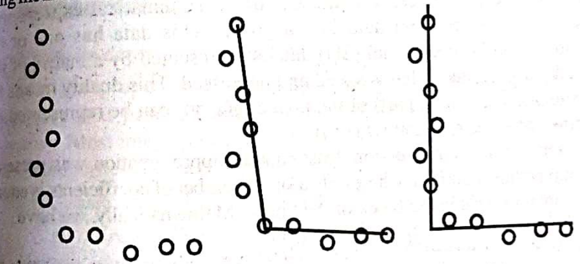
This algorithm does not provide guarantee that a path find by it is minimum-cost path.

Q.42. Explain the term line fitting.

Ans. A simple way to fit a straight line to a curve is just to specify the endpoints of the curve as the straight line endpoints. On-valued pixels are shown in fig. 2.41 (a) and two corner fits are shown in fig. 2.41(b) and (c). In fig. 2.41 (b), the fit is made exactly on data points. Because of this, the endpoints are sure to connect, but the corner angle is not as sharp as intended in the original. In fig. 2.41 (c), a least squares fit was performed to two sets of points, and the intersection of them was taken as the corner location. This yields a sharper corner. However, this may result in some portions of the curve having large error with respect to the fit. A popular way to achieve the lowest average error for all points on the curve is to perform a least-squares fit of a line to the points

on the curve. For a line fit of equation, y = mx + c, the objective is to determine m and c from the data points, (x_i, y_i). The least square fit is made by minimizing the sum of errors, $\Sigma(y - y_i)^2$ over all data points, i = 1, ..., n. The solution is found by solving the simultaneous equations, $\Sigma y_i = m\Sigma x_i + cn$, and $\Sigma x_i y_i = m\Sigma x_i^2 + c\Sigma x_i$, for all data points, i, to obtain m and c.

There can be a problem in blindly applying the least-squares method just described to document analysis applications. That is, the method is not independent of orientation, and as the true slope of the line approaches vertical, this method of minimizing y-error becomes inappropriate. One approach is to minimize y-error only for slopes expected to be ±45 degrees around the horizontal and to minimize x-error for slopes expected to be ± 45 degrees around the vertical. A more general approach is to minimize error not with respect to x or y, but with respect to the perpendicular distance to the line in any orientation. This can be done by a line-fitting method called principal axis determination or eigenvector line fitting.



(a) (b) (c)

Fig. 2.41

Example – Using Hough transform show that the points (1,1), (2, 2), and (3, 3) are collinear and find the equation of line.

We know that, the equation of line is y = mx + c

In order to perform Hough transform we need to convert line from (x, y) plane to (m, c) plane.

Equation of (m, c) plane is

Step 1 –

y = mx + c

For (1, 1)

1 = m + c

c = – m + 1

If c = 0 then (0 = – m + 1) m = 1

If m = 0 then (c = 0 + 1) c = 1

(m, c) = (1, 1)

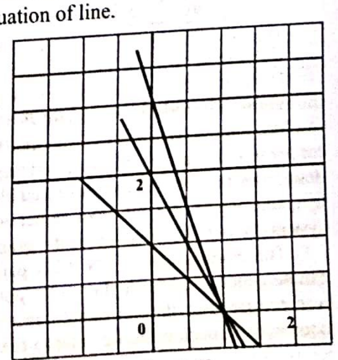


Fig. 2.42

Similarly for

If (x,y) = (2,2) then (m, c) = (1,2)

If (x,y) = (3,3) then (m, c) = (1,3)

Step 2 –

Plot a graph for (m, c) = (1, 1), (1, 2) and (1, 3)

Step 3 – The original equation of line is (y = mx + c) put the value of m and c on this equation.

Then y = x

points (1,1), (2,2), and (3,3) are collinear.

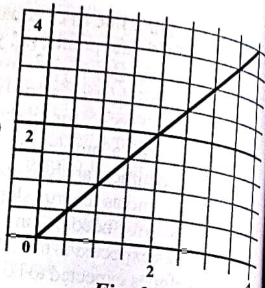


Fig. 2.43

Q.43. Describe the curve fitting of image.

Ans. Curve-fitting refers to the use of a mathematical expression to represent gathered random data into a group. This data has one or two dimensions. The 2-dimensional (2D) data is represented by a matrix of data such as the digital image that is saved and processed. This duality means that the image or block (small part) of the image (g(x, y)) can be represented as a 2D mathematical expression (z(x, y)).

To achieve data compression, least squares approximation was presented as an optimization algorithm where only a small number of coefficients is enough to represent all pixels in the block of the image. Mathematically, we have

$$\min \sum_{x}\sum_{y}[z(x,y)-g(x,y)]^2$$

where g(x, y) is the original intensity value of the image (or any color component) and z(x, y) is the value from the suggested function. A simplified derivation of first order (plane) fitting was proposed as

$$\min_{a,b,c} \sum_{x}\sum_{y}[ax+by+c-g(x,y)]^2$$

The coefficients a, b and c of a 2n × 2N block are computed from their N × N counterparts and were assumed uniformly distributed. The resulting values of the minimization procedure, usually 8 × 8 or 16 × 16, were retained whenever the resulting error was less than a prescribed threshold. A PSNR of 32 dB was reported for 16:1 compression (0.5 bpp) with high complexity in building the quad tree describing the sizes of the compressed blocks. To reduce the error energy imposed by quantizing a and b, the block center was selected as the origin of the coordinate system. In fact, the selection of the origin can also affect the range values of c.

The computation of 2N × 2N parameters from their N × N counterparts can be generalized for higher-order polynomials. A related quad tree approach was proposed to predict block corners from the upper left one. These four corners were used in the decoder to find the coefficients of (dxy + ax + by + c).

UNIT 3

REGION ANALYSIS – REGION PROPERTIES, EXTREMAL POINTS, SPATIAL MOMENTS, MIXED SPATIAL GRAY-LEVEL MOMENTS

Q.1. Write down some basic properties of region.

Ans. The properties of the regions become the input to higher level procedures that perform decision-making tasks such as recognition or inspection. Most image processing packages have operators that can produce a set of properties for each region.

Some basic properties of regions are as follows –

(i) Area of Region – We denote the set of pixels in a region by R_E. The simplest geometric properties are the region's area denoted by A and the centroid denoted by $(\overline{r_e},\overline{c_e})$. The area of region defined as –

$$A = \sum_{(r_e,c_e)\in R_E} 1$$

It means the area is just a count of the pixels in the region R_E.

(ii) Centroid of Region – The centroid of region $(\overline{r_e},\overline{c_e})$ is thus the 'average' location of the pixels in the set R_E. The centroid is defined as –

$$\overline{r_e} = \frac{1}{A}\sum_{(r_e,c_e)\in R_E} \overline{r_e}$$

$$\overline{c_e} = \frac{1}{A}\sum_{(\overline{r_e},\overline{c_e})\in R_E} \overline{ce}$$

Note that, even though $(r_e, c_e) \in R_E$ is a pair of integers $(\overline{r_e},\overline{c_e})$ is generally not a pair of integers, often a precision of tenths of a pixel is justifiable for the centroid.

(iii) Perimeter of Region – A simple definition of the perimeter of a region without holes is the set of its interior border pixels. Perimeter of region is denoted by P_E, and a pixel of a region is a border pixel if it has some neighbouring pixel that is outside the region. For example, the resulting set of perimeter pixels will be 4-connected, when 8-connectivity is used to determine whether a pixel inside the region is connected to a pixel outside the region. Similarly, the resulting set of perimeter pixels well be 8-connected, when 4-connectivity is used to determine whether a pixel inside the region is connected to a pixel outside the region.

The 4-connected perimeter P_{E_4} and the 8-connected perimeter P_{E_8} are defined as –

$$P_{E_4} = \{(r_e, c_e) \in R_E | N_8 (r_e, c_e) - R_E \neq \phi\}$$

$$P_{E_8} = \{(r_e, c_e) \in R_E | N_4 (r_e, c_e) - R_E \neq \phi\}$$

Here N is the number of connections.

(iv) The Perimeter Length of Region – To compute length $|P_E|$ of perimeter P_E, the pixels in P_E must be ordered in a sequence $P_E = <(r_{e_0}, c_{e_0}), \ldots, (r_{e_{M-1}}, c_{e_{M-1}})>$, each pair of successive pixels in the sequence being neighbours, including the first and last pixels. Then the perimeter length $|P_E|$ is defined as –

$$\begin{aligned}|P_E| = &| \{m \mid (r_{e_{m+1}}, c_{e_{m+1}}) \in N_4(r_{e_m}, c_{e_m})\} | \\ &+ \sqrt{2} \,| \{m \mid (r_{e_{m+1}}, c_{e_{m+1}}) \in N_8(r_{e_m}, c_{e_m}) \\ &- N_4(r_{e_m}, c_{e_m})\} |\end{aligned}$$

Here, m+1 is computed modulo M, the length of the pixel sequence.

(v) The First Circularity Measure of Region – A common measure of circularity of the region is the length of the perimeter squared divided by the area. The first circularity is defined as –

$$C_1 = \frac{|P_E|^2}{A}$$

However, for digital shapes, $|P_E|^2/A$ assumes its smallest value not for digital circles, as it would for continuous planar shapes, but for digital octagons or diamonds depending on whether the perimeter is computed as the number of its 4-neighbouring border pixels or as the length of the border, counting 1 for vertical or horizontal moves and $\sqrt{2}$ for diagonal moves.

(vi) The Second Circularity Measure of Region – Haralick proposed a second circularity measure in 1974 to overcome the first circularity measure problem. The second circularity is defined as –

$$C_2 = \frac{\mu_{R_E}}{\sigma_{R_E}}$$

Here, μ_{R_E}, σ_{R_E} are the mean and standard deviation of the distance from the centroid of the shape to the shape boundary. Circularity measure can be computed by mean radial distance and standard deviation of radial distance formulae. They are defined below –

Mean Radial Distance – The mean radial distance is defined as –

$$\mu_{R_E} = \frac{1}{M} \sum_{m=0}^{M-1} \| (r_{e_m}, c_{e_m}) - (\bar{r}_e - \bar{c}_e) \|$$

Standard Deviation of Radial Distance – The standard deviation of radial distance is defined as –

$$\sigma_{R_E} = \sqrt{\frac{1}{M} \sum_{m=0}^{M-1} [\| (r_{e_m}, c_{e_m}) - (\bar{r}_e - \bar{c}_e) \| - \mu_{R_E}]^2}$$

Here, the set of pixels (r_{e_m}, c_{e_m}), $m = 0, \ldots, M-1$ lie on the perimeter P_E of the region. The second circularity measure C_2 increases monotonically as the digital shape becomes more circular and is similar for digital and continuous shapes.

Q.2. Explain the term extremal points in region properties.

Ans. Extremal points is the main properties of region. A bounding box is a rectangle with horizontal and vertical sides that encloses the region and touches its topmost, bottommost, leftmost, and rightmost points. In fig. 3.1, there can be as many as eight distinct extremal pixels to a region – topmost right (TR), rightmost top (RT), rightmost bottom (RB), bottommost right (BR), bottommost left (BL), leftmost bottom (LB), leftmost top (LT) and topmost left (TL). Each extremal point has a extremal coordinate value in either its row or column coordinate position. Each extremal point lies on the bounding box of the region. Extremal points occur in opposite pairs such as TL with BR, TR with BL, RT with LB and RB with LT. Each pair of opposite extremal points defines an axis. Useful properties of the axis include its axis length and orientation. Because the extremal points come from a spatial digitization or quantization, the standard Euclidean distance formula will provide distances that are biased slightly low.

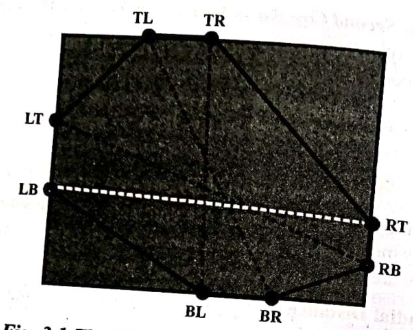


Fig. 3.1 The Eight Extremal Points of Region

For example, let consider the length covered by two pixels horizontally adjacent. From the left edge of the left pixel to the right edge of the right pixel is a length of 2 but the distance between the pixel centers is only 1. The appropriate calculation for distance adds a small increment to the Euclidean distance to account for this. The increment depends on the orientation angle θ of the axis and is given by

$$X(\theta) = \begin{cases} \dfrac{1}{|\cos\theta|} & : |\theta| < 45^\circ \\ \dfrac{1}{|\sin\theta|} & : |\theta| > 45^\circ \end{cases}$$

with this increment, the length of the extremal axis from extremal point (r_{e_1}, c_{e_1}) to extremal point (r_{e_2}, c_{e_2}) is defined as –

Extremal axis length –

$$Y = \left[(r_{e_2} - r_{e_1})^2 + (c_{e_2} - c_{e_1})^2\right]^{1/2} + X(\theta)$$

Q.3. Write short note on spatial moments region properties.

Ans. Another important region property is spatial moments. Spatial moments are often used to describe the shape of a region. There are three 2nd order spatial moments of a region. They are denoted by $\mu_{(r_e,r_e)}, \mu_{(r_e,c_e)}$ and $\mu_{(c_e,c_e)}$, where $\mu_{(r_e,r_e)}$ measures row variation from the row mean, $\mu_{(c_e,c_e)}$ measures column variation from the column mean and $\mu_{(r_e,c_e)}$ measures row and column variation from the centroid.

The 2nd order row moment is defined as –

$$\mu_{(r_e,c_e)} = \frac{1}{A} \sum_{(r_e,c_e)\in R_E} (r_e - \bar{r}_e)^2$$

The 2nd order column moment is defined as –

$$\mu_{(c_e,c_e)} = \frac{1}{A} \sum_{(r_e,c_e)\in R_E} (c_e - \bar{c}_e)^2$$

The 2nd order row column (mixed) moment is defined as –

$$\mu_{(r_e,c_e)} = \frac{1}{A} \sum_{(r_e,c_e)\in R_E} (r_e - \bar{r}_e)(c_e - \bar{c}_e)$$

These quantities are often used as simple shape descriptors, as they are invariant to translation and scale change of a 2D shape. The 2nd order spatial moments have value and meaning for a region of any shape, the same way that the covariance matrix has value and meaning for any two-dimensional probability distribution.

Q.4. Discuss about the mixed spatial gray level moments region properties.

Ans. A simple gray level properties consist gray level mean and variance. Other gray level properties consist the mixed spatial gray level moments. There are two 2nd order mixed gray level spatial moments which are defined as –

(i) $$\mu_{(r_e,g)} = \frac{1}{A} \sum_{(r_e,c_e)\in R_E} (r_e - \bar{r}_e)[I(r_e - c_e) - \mu]$$

(ii) $$\mu_{(c_e,g)} = \frac{1}{A} \sum_{(r_e,c_e)\in R_E} (c_e - \bar{c}_e)[I(r_e - c_e) - \mu]$$

These spatial moments can be used to determine the least squares, best fit gray level intensity planes to the observed gray level spatial pattern of the region R_E. The least-squares fit to the observed $I(r_e, c_e)$ is the gray level intensity plane $X(r_e - \bar{r}_e) + Y(c_e - \bar{c}_e) + Z$ determined from the X, Y and Z that minimizes, defined by –

$$\epsilon^2 = \sum_{(r_e,c_e)\in R_E} \left[X(r_e - \bar{r}_e) + Y(c_e - \bar{c}_e) + Z - I(r_e, c_e)\right]^2$$

Taking partial derivatives of ϵ^2 with respect to X, Y and Z and setting these partial derivatives to zero leads to the normal regression equation that in

this instance is defined as –

$$\begin{bmatrix} \sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)^2 & \sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)(c_e-\overline{c}_e) & \sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e) \\ \sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)(c_e-\overline{c}_e) & \sum\limits_{(r_e,c_e)\in R_E}(c_e-\overline{c}_e)^2 & \sum\limits_{(r_e,c_e)\in R_E}(c_e-\overline{c}_e) \\ \sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e) & \sum\limits_{(r_e,c_e)\in R_E}(c_e-\overline{c}_e) & \sum\limits_{(r_e,c_e)\in R_E}1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)I(r_e,c_e) \\ \sum\limits_{(r_e,c_e)\in R_E}(c_e-\overline{c}_e)I(r_e,c_e) \\ \sum\limits_{(r_e,c_e)\in R_E}I(r_e,c_e) \end{bmatrix} \quad ...(i)$$

Since $\sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)=0$ and $\sum\limits_{(r_e,c_e)\in R_E}(c_e-\overline{c}_e)=0$, put these values in equation (i), then equation will be –

$$\begin{bmatrix} \sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)^2 & \sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)(c_e-\overline{c}_e) & 0 \\ \sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)(c_e-\overline{c}_e) & \sum\limits_{(r_e,c_e)\in R_E}(c_e-\overline{c}_e)^2 & 0 \\ 0 & 0 & \sum\limits_{(r_e,c_e)\in R_E}1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \sum\limits_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)(I(r_e,c_e)-Z) \\ \sum\limits_{(r_e,c_e)\in R_E}(c_e-\overline{c}_e)(I(r_e-c_e)-Z) \\ \sum\limits_{(r_e,c_e)\in R_E}I(r_e-c_e) \end{bmatrix} \quad ...(ii)$$

Hence, $Z=\frac{1}{A}\sum\limits_{(r_e,c_e)\in R_E}I(r_e-c_e)=\mu$

Recalling that, 2nd order spatial moments of regions –

$$\mu_{(r_e,r_e)}=\frac{1}{A}\sum_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)^2$$

$$\mu_{(c_e,c_e)}=\frac{1}{A}\sum_{(r_e,c_e)\in R_E}(c_e-\overline{c}_e)^2$$

$$\mu_{(r_e,c_e)}=\frac{1}{A}\sum_{(r_e,c_e)\in R_E}(r_e-\overline{r}_e)(c_e-\overline{c}_e)$$

We know that the unknown parameters X and Y must satisfy –

$$\begin{bmatrix} \mu_{(r_e,r_e)} & \mu_{(r_e,c_e)} \\ \mu_{(r_e,c_e)} & \mu_{(c_e,c_e)} \end{bmatrix}\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \mu_{(r_e,g)} \\ \mu_{(c_e,g)} \end{bmatrix}$$

According to Kramer's rule, solving for X and Y, we get

$$X=\frac{\begin{bmatrix} \mu_{(r_e,g)} & \mu_{(r_e,c_e)} \\ \mu_{(c_e,g)} & \mu_{(c_e,c_e)} \end{bmatrix}}{\begin{bmatrix} \mu_{(r_e,r_e)} & \mu_{(r_e,c_e)} \\ \mu_{(r_e,c_e)} & \mu_{(c_e,c_e)} \end{bmatrix}} \qquad Y=\frac{\begin{bmatrix} \mu_{(r_e,r_e)} & \mu_{(r_e,g)} \\ \mu_{(r_e,c_e)} & \mu_{(c_e,g)} \end{bmatrix}}{\begin{bmatrix} \mu_{(r_e,r_e)} & \mu_{(r_e,c_e)} \\ \mu_{(r_e,c_e)} & \mu_{(c_e,c_e)} \end{bmatrix}}$$

Therefore the equation of the fitted plane is defined by –

$$\hat{I}(r_e,c_e) = X(r_e-\overline{r}_e)+Y(c_e-\overline{c}_e)+\mu,\ (r_e,c_e)\in R_E$$

BOUNDARY ANALYSIS – SIGNATURE PROPERTIES, SHAPE NUMBERS

Q.5. Discuss about some signature properties.

Ans. Signature properties obtainable from vertical projection, horizontal projection and diagonal projection, include area, centroid of region, second moments and bounding rectangle. These are as follows –

(i) Vertical Projection – We denote the vertical projection by V_P. It is defined as –

$$V_P(c_e) = \{r_e \mid (r_e, c_e) \in R_E\}$$

(ii) Horizontal Projection – We denote the horizontal projection by H_P. It is defined as –

$$H_P(r_e) = \{c_e \mid (r_e, c_e) \in R_E\}$$

(iii) Diagonal Projection – There are two diagonal projections. First going from lower left to upper right and second, going from upper left to lower right. We denote diagonal projection by D_P. The diagonal projection D_{P_L} goes from lower left to upper right, is defined by

$$D_{P_L}(l) = \#\{(r_e, c_e) \in R_E \mid r_e + c_e = l\}$$

The second diagonal projection D_{P_U} goes from upper right to lower left, is defined by

$$D_{P_U}(u) = \#\{(r_e, c_e) \in R_E \mid r_e - c_e = u\}$$

(iv) Area of Region – The area A can be obtained from any projection. The area A is defined as –

$$A = \sum_{(r_e,c_e)\in R_E} 1 = \sum_{r_e} \sum_{\{c_e \mid (r_e,c_e)\in R_E\}} 1 = \sum_{r_e} H_P(r_e)$$

(v) Bounding Rectangle – They are divided in four sections, top row, bottom row, leftmost and rightmost.

(a) Top Row – It is denoted by r_{min}. The top row of bounding rectangle is given by

$$r_{min} = \min\{r_e \mid (r_e, c_e) \in R_E\}$$

$$= \min\{r_e \mid H_P(r_e) \neq 0\}$$

(b) Bottom Row – This can be denoted by r_{max}. The bottom row of the bounding rectangle rectangle is given by

$$r_{max} = \max\{r_e \mid (r_e, c_e) \in R_E\}$$

$$= \max\{r_e\,(H_P(r_e) \neq 0\}$$

(c) Leftmost – This can be denoted by c_{min}. The leftmost column of the bounding rectangle is given by

$$c_{min} = \min\{c_e \mid (r_e, c_e) \in R_E\}$$

$$= \min\{c_e \mid V_P(c_e) \neq 0\}$$

(d) Rightmost – This can be denoted by c_{max}. The rightmost column of the bounding rectangle is given by

$$c_{max} = \max\{c_e \mid (r_e, c_e) \in R_E\}$$

$$\max\{c_e \mid V_P(c_e) \neq 0\}$$

(vi) Centroid of Region – The centroid of region include horizontal projection, vertical projection, diagonal projection. These are as follows –

(a) Row Centroid – The row centroid $\overline{r_e}$ can be obtained from the horizontal projection H_P, as defined below –

$$\overline{r_e} = \frac{1}{A} \sum_{(r_e,c_e)\in R_E} r_e$$

$$= \frac{1}{A} \sum_{r_e} \sum_{\{c_e \mid (r_e,c_e)\in R_E\}} r_e$$

$$= \frac{1}{A} \sum_{r_e} r_e \sum_{\{c_e \mid (r_e,c_e)\in R_E\}} 1$$

$$= \frac{1}{A} \sum_{r_e} r_e H_P(r_e)$$

(b) Column Centroid – The column centroid $\overline{c_e}$ can be obtained from the vertical projection V_P as follows –

$$\overline{c_e} = \frac{1}{A} \sum_{(r_e,c_e)\in R_E} c_e$$

$$= \frac{1}{A} \sum_{c_e} \sum_{\{r_e \mid (r_e,c_e)\in R_E\}} c_e$$

$$= \frac{1}{A} \sum_{c_e} c_e \sum_{\{c_e \mid (r_e,c_e)\in R_E\}} 1$$

$$= \frac{1}{A} \sum_{c_e} c_e V_P(c_e)$$

(c) Diagonal Centroid (Lower Left to Upper Right) – The diagonal centroid $\bar{l}$ can be obtained from the diagonal projection D_{P_L}.

$$\bar{l} = \frac{1}{A} \sum_{l} l\, D_{P_L}(l)$$

The diagonal centroid $\bar{l}$ is related to the row and column centroid in following manner –

$$\bar{l} = \frac{1}{A} \sum_{l} l \sum_{\{(r_e,c_e)\in R_E \mid r_e + c_e = l\}} 1$$

$$= \frac{1}{A} \sum_{l} \sum_{\{(r_e,c_e)\in R_E \mid r_e + c_e = l\}} (r_e + c_e)$$

$$= \frac{1}{A}\sum_{l} \sum_{\{(r_e,c_e)\in R_E | r_e+c_e=l\}} r_e$$

$$+\frac{1}{A}\sum_{l} \sum_{\{(r_e,c_e)\in R_E | r_e+c_e=l\}} c_e$$

$$= \frac{1}{A}\sum_{(r_e,c_e)\in R_E} r_e + \frac{1}{A}\sum_{(r_e,c_e)\in R_E} c_e$$

$$= \overline{r_e} + \overline{c_e}$$

(d) Diagonal Centroid (Upper Left to Lower Right) – The diagonal centroid $\overline{U}$ can be obtained from the diagonal projection D_{P_U}.

$$\overline{U} = \frac{1}{A}\sum_{u} u\, D_{P_U}(u)$$

Similarly, the diagonal centroid u is related to the row and column centroid in following way –

$$\overline{u} = \overline{r_e} - \overline{c_e}$$

(vii) Second Moments – Second moments also include horizontal projection, vertical projection, diagonal projection.

(a) Row Moment – The second row moment $\mu_{(r_e,r_e)}$ can be obtained from the horizontal projection H_P, defined as –

$$\mu_{(r_e,r_e)} = \frac{1}{A}\sum_{(r_e,c_e)\in R_E} (r_e - \overline{r_e})^2$$

$$= \frac{1}{A}\sum_{r_e} \sum_{\{c_e|(r_e,c_e)\in R_E\}} (r_e - \overline{r_e})^2$$

$$= \frac{1}{A}\sum_{r_e} (r_e - \overline{r_e})^2 \sum_{\{c_e|(r_e,c_e)\in R_E\}} 1$$

$$= \frac{1}{A}\sum_{r_e} (r_e - \overline{r_e})^2 H_P(r_e)$$

(b) Column Moment – The second column moment $\mu_{(c_e,c_e)}$ can be obtained from the vertical projection V_P, defined as –

$$\mu_{(c_e,c_e)} = \frac{1}{A}\sum_{(r_e,c_e)\in R_E} (c_e - \overline{c_e})^2$$

$$= \frac{1}{A}\sum_{c_e} \sum_{\{r_e|(r_e,c_e)\in R_E\}} (c_e - \overline{c_e})^2$$

$$= \frac{1}{A}\sum_{c_e} (c_e - \overline{c_e})^2 \sum_{\{r_e|(r_e,c_e)\in R_E\}} 1$$

$$= \frac{1}{A}\sum_{c_e} (c_e - \overline{c_e})^2 V_P(c_e)$$

(c) Diagonal Moment (Lower Left to Upper Right) – The second diagonal moment $\mu_{(l,l)}$ can be obtained from the diagonal projection D_{P_L}, defined as –

$$\mu_{(l,l)} = \frac{1}{A}\sum_{l} (l - \overline{l})^2 D_{P_L}(l)$$

The second diagonal moment $\mu_{(l,l)}$ is related to $\mu_{(r_e,c_e)}, \mu_{(r_e,r_e)}$ and $\mu_{(c_e,c_e)}$ in following way

$$\mu_{(l,l)} = \frac{1}{A}\sum_{l} \sum_{\{(r_e,c_e)\in R_E | r_e+c_e=d\}} (r_e + c_e - \overline{r_e} - \overline{c_e})^2$$

$$= \frac{1}{A}\sum_{(r_e,c_e)\in R_E} [(r_e - \overline{r_e}) - (c_e - \overline{c_e})]^2$$

$$= \frac{1}{A}\sum_{(r_e,c_e)\in R_E} (r_e - \overline{r_e})^2 + 2(r_e - \overline{r_e})(c_e - \overline{c_e}) + (c_e - \overline{c_e})^2$$

$$= \mu_{(r_e,r_e)} + 2\mu_{(r_e,c_e)} + \mu_{(c_e,c_e)}$$

Hence, the second mixed moment can be obtained from the second diagonal moment $\mu_{(l,l)}$ by

$$\mu_{(r_e,c_e)} = \frac{\mu_{(l,l)} - \mu_{(r_e,r_e)} - \mu_{(c_e,c_e)}}{2}$$

Scanned with CamScanner

(d) Diagonal Moment (Upper Left to Lower Right) – The second diagonal moment $\mu_{(u,u)}$ is also related to $\mu_{(r_e,c_e)}$, $\mu_{(r_e,r_e)}$ and $\mu_{(c_e,c_e)}$.

$$\mu_{(u,u)} = \frac{1}{A}\sum_{u}\sum_{\{(r_e,c_e)\in R_E \mid r_e - c_e = u\}}(r_e - c_e - \bar{r}_e + \bar{c}_e)^2$$

$$= \frac{1}{A}\sum_{(r_e,c_e)\in RE}[(r_e - \bar{r}_e) - (c_e - \bar{c}_e)]^2$$

$$= \frac{1}{A}\sum_{(r_e,c_e)\in R_E}(r_e - \bar{r}_e)^2 - 2(r_e - \bar{r}_e)(c_e - \bar{c}_e) + (c_e - \bar{c}_e)^2$$

$$= \mu_{(r_e,r_e)} - 2\mu_{(r_e,c_e)} + \mu_{(c_e,c_e)}$$

Hence, the second mixed moment can also be obtained from the second diagonal moment $\mu_{(u,u)}$ by

$$\mu_{(r_e,c_e)} = \frac{\mu_{(r_e,r_e)} + \mu_{(c_e,c_e)} - \mu_{(u,u)}}{2}$$

The relationship between the two diagonal moments $\mu_{(l,l)}$ and $\mu_{(l,l)}$ implies that the mixed moment $\mu_{(r_e,c_e)}$ can be obtained directly from $\mu_{(l,l)}$ and $\mu_{(u,u)}$.

$$\mu_{(r_e,c_e)} = \frac{\mu_{(l,l)} - \mu_{(u,u)}}{4}$$

Q.6. Explain the term shape number with example.

Ans. The shape number of a chain-coded boundary, based on the 4-directional code as shown in fig. 3.2, is defined as the first difference of smallest magnitude. The order n of a shape number is defined as the number of digits in its representation. Moreover, n is even for a closed boundary and its value limits the number of possible different shapes. All the shapes of order 4 and 6, along with their chain-code representations, I[st] differences, and corresponding shape numbers are shown in fig. 3.2 (b). The shape number follows from the I[st] difference of this code. Although the order of the resulting shape number usually equals n because of the way the grid spacing was selected, boundaries with depressions comparable to this spacing sometimes yield shape numbers of order greater than n. In this case, we specify a rectangle of order lower than n and repeat the procedure until the resulting shape number is of order n. For example, Assume n = 18 is specified for the boundary shown in fig. 3.2 (c). The I[st] step is to find the basic rectangle as shown in fig. 3.2 (d). The closest rectangle of order 18 is a 3 × 6 rectangle, requiring subdivision of the basic rectangle as shown in fig. 3.2 (e), where the chain-code directions are aligned with the resulting grid. In fig. 3.2 (f), the last step is to obtain the chain code and use its I[st] difference to compute the shape number.

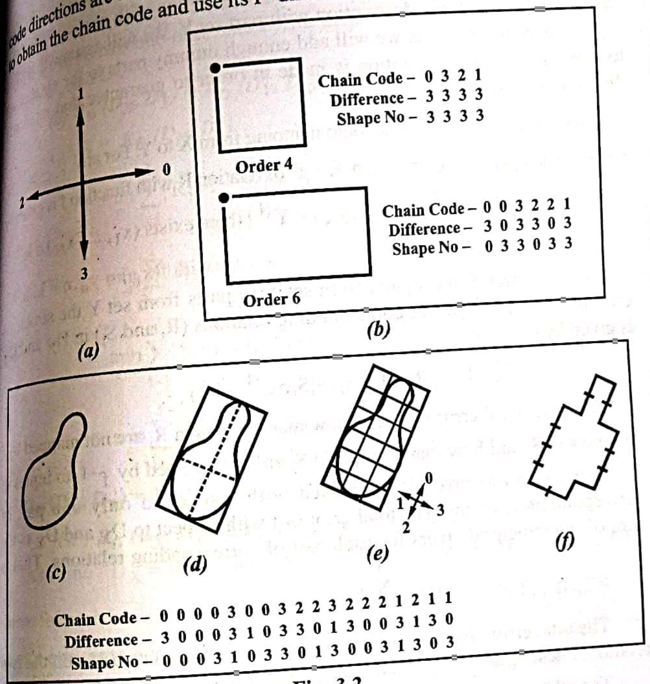


Fig. 3.2

GENERAL FRAMEWORKS FOR MATCHING – DISTANCE RELATIONAL APPROACH, ORDERED STRUCTURAL MATCHING, VIEW CLASS MATCHING, MODELS DATABASE ORGANIZATION

Q.7. Discuss about the relational distance as a framework for matching.

Ans. A relational description D_A is a sequence of relations $D_A = \{R_1, ..., R_I\}$, where for each i = 1, ..., I, there exists a positive integer n_i with $R_i \subseteq A^{n_i}$ for some set A. Intuitively A is a set of the parts of the entity being described, and the relation R_i indicate various relationships among the parts. A relational description is a data structure that may be used to describe two dimensional

shape models, three dimensional object models, regions on an image, and so on. Let $D_X = \{R_1, ..., R_I\}$ be a relational description with part set X and $D_Y = \{S_1, ... S_I\}$ a relational description with part set Y. We will assume that $|X| = |Y|$; if this is not the case, we will add enough dummy parts to the smaller set to make it so. The assumption is made in order to guarantee that relational distance is a metric.

Assume, f be any one-one, onto mapping from X to Y. For any $R \subseteq X^N$, N a positive integer, the composition $R \circ f$ of relation R with function f is given by

$$R \circ f = \{(y_1,....., y_N) \in Y^N \mid \text{there exists } (x_1,....., x_N) \in R \text{ with } f(x_n) = y_n, n = 1,...., N\}$$

The function f maps parts from set X to parts from set Y the structural error of f for the i^{th} pair of corresponding relations (R_i and S_i) in D_X and D_Y is given by

$$E_s^i(f) = |R_i \circ f - S_i| + |S_i \circ f^{-1} - R_i|$$

The structural error indicates how many tuples in R_i are not mapped by f to tuples in S_i and how many tuples in S_i are not mapped by f^{-1} to tuples in R_i. The structural error is expressed with respect to only one pair of corresponding relations. The total error of f with respect to D_X and D_Y is the sum of the structural errors for each pair of corresponding relations. That is

$$E(f) = \sum_{i=1}^{I} E_s^i(f)$$

The total error gives a quantitative idea of the difference between the two relational descriptions D_X and D_Y with respect to the mapping f.

The relational distance $R_D(D_X, D_Y)$ between D_X and D_Y is then denoted by

$$R_D(D_X, D_Y) = \min_{\substack{1-1 \\ f:X \to Y \\ \text{onto}}} E(f)$$

It means that, the relational distance is the minimal total error obtained for any one-one, onto mapping f from X to Y. We call a mapping f that minimizes total error a best mapping from D_X to D_Y. If there is more than one best mapping, one can be arbitrarily selected as the designated best mapping.

Q.8. Suppose R_D be the relational-distance measure, and let D_X, D_Y and D_Z be arbitrary relational descriptions, then prove that

(i) $R_D(D_X, D_Y) = 0$, if and only if D_X and D_Y are isomorphic

(ii) $R_D(D_X, D_Y) = R_D(D_Y, D_X)$

(iii) $R_D(D_X, D_Y) = R_D(D_X, D_Z) + R_D(D_Z, D_Y)$.

Ans. (i) $R_D(D_X, D_Y) = 0$

If f is an isomorphism between D_X and D_Y, then $E(f) = 0$. If $R_D(D_X, D_Y) = 0$, then there exists one-one, onto f with $E(f) = 0$. Thus f is an isomorphism between D_X and D_Y.

(ii) $R_D(D_X, D_Y) = R_D(D_Y, D_X)$

L.H.S. = $R_D(D_X, D_Y)$

$$= \min_f \sum_{i=1}^{I} |R_i \circ f - S_i| + |S_i \circ f^{-1} - R_i|$$

$$= \min_{f^{-1}} \sum_{i=1}^{I} |R_i \circ f^{-1} - S_i| + |S_i \circ (f^{-1}) - R_i|$$

$$= \min_{f^{-1}} \sum_{i=1}^{I} |S_i \circ f - R_i| + |R_i \circ f^{-1} - S_i|$$

$$= \min_f \sum_{i=1}^{I} |S_i \circ f - R_i| + |R_i \circ f^{-1} - S_i|$$

$R_D(D_Y, D_X)$ = **R.H.S.**

(iii) $R_D(D_X, D_Y) \le R_D(D_X, D_Z) + R_D(D_Z, D_Y)$

Assume $D_X = \{R_1,.....R_I\}$, $D_Y = \{S_1,.....S_I\}$ and $D_Z = \{T_1,.....T_I\}$, where for each i = 1, ..., I, $R_i \subseteq X^{n_i}$, $S_i \subseteq Y^{n_i}$ and $T_i \subseteq Z^{n_i}$. Let $f_1 \subseteq X \times Z$ be one-one, onto and the f_1 that minimizes $R_D(D_X, D_Z)$. Let $f_2 \subseteq Z \times Y$ be one-one, single valued and the f_2 that minimizes $R_D(D_Z, D_Y)$.

Let $f: X \to Y = f_1 \circ f_2$. Then f is one-one and onto and produces some error E(f) with respect to D_X and D_Y. Assume $x \in R_i \circ f_1 \circ f_2 - S_i$. Then $x \in R_i \circ f_1 \circ f_2$ and $x \notin S_i$. Since $x \in R_i \circ f_1 \circ f_2$ and f_2 is one-one and onto, there exists a unique $y \in R_i \circ f_1$ such that $\{y\} = \{x\} \circ f_2^{-1}$ and $\{x\} = \{y\} \circ f_2$. If $y \notin T_i$, then $y \in R_i \circ f_1 - T_i$. If $y \in T_i$, then $\{x\} = \{y\} \circ f_2$ is an element of $T_i \circ f_2$. Hence $x \in T_i \circ f_2 - S_i$.

Since for each $x \in R_i \circ f_1 \circ f_2 - S_i$ either $x \in T_i \circ f_1 \circ f_2 - S_i$ or $y = x \circ f_2^{-1} \in R_i \circ f_1 - T_i$, we have

$$|R_i \circ f_1 \circ f_2 - S_i| \le |R_i \circ f_1 - T_i| + |T_i \circ f_2 - S_i|$$

Thus

$$\sum_{i=1}^{I}|R_i \circ f_1 \circ f_2 - S_i| \le \sum_{i=1}^{I}|R_i \circ f_1 - T_i| + \sum_{i=1}^{I}|T_i \circ f_2 - S_i|$$

Similarly, we can show that

$$\sum_{i=1}^{I}|S_i \circ f_2^{-1} \circ f_1^{-1} - S_i| \le \sum_{i=1}^{I}|T_i \circ f_1^{-1} - R_i| + \sum_{i=1}^{I}|S_i \circ f_2^{-1} - T_i|$$

Adding, we get

$$\sum_{i=1}^{I}|R_i \circ f_1 \circ f_2 - S_i| + |S_i \circ f_2^{-1} \circ f_1^{-1} - S_i|$$

$$\le \sum_{i=1}^{I}|R_i \circ f_1 - T_i| + |T_i \circ f_1^{-1} - R_i| + \sum_{i=1}^{I}|T_i \circ f_2 - S_i| + |S_i \circ f_2^{-1} - T_i|$$

Which says

$$E(f) \text{ wrt } D_X \text{ and } D_Y \le R_D(D_X, D_Z) + R_D(D_Z, D_Y)$$

But $\quad R_D(D_X, D_Y) \le E(f)$

so $\quad R_D(D_X, D_Y) \le R_D(D_X, D_Z) + R_D(D_Z, D_Y)$

Thus the relational distance of two relation description is a metric up to isomorphism.

Q.9. Write short note on ordered structural matching.

Ans. In 1987, ordered structural matching was used by Shapiro, MacDonald and Sternberg, for 2-D shape matching using shape primitives extracted by operations of mathematical morphology. Another example of ordered structural matching are syntactic pattern recognition algorithms that represent object models by grammars, that convert an object to a string of symbols and that parse the string according to the grammar.

In many two-dimensional computer vision problems, the spatial arrangement of the primitives allows the definition of an ordering on the primitives that greatly reduces the complexity of the search.

Assume that we wish to compare an object represented by description O_X to another object represented by description O_Y. Suppose that the primitive set X has the ordering $<x_1, x_2, ..., x_s>$ and that the primitive set Y has the ordering $< y_1, y_2, ..., y_t >$. Suppose that during the matching process it is hypothesized that primitive x_i maps to primitive y_j. The ordering tells us that one of the following conditions holds –

(i) x_{i+1} maps to y_{j+1}.

(ii) x_{i+1} maps to $y_{j+k}, k > 1$, and no primitive of O_X maps to any of $y_{j+1}, y_{j+2}, ..., y_{j+k-1}$.

(iii) x_{i+1} maps to no primitive of O_Y.

here, x_{i+1} is the next primitive after x_i in the ordering. Thus, once x_i is mapped to y_j, the ordering can be used to find the correspondences between all the other primitives in polynomial time.

Q.10. Explain various stages of view class matching.

Ans. A view class matching can be divided into two stages, when a 3-D object is represented by a view-class model. These stages are as follows –

(i) Determining View Class of the Object – A view class is represented by a hierarchical relational structure called the relational pyramid, in which primitives appear on the lowest level and each of the other levels represents relationships among entities from lower levels. The full relational pyramid structure is for use in detailed matching for determining the exact pose of the object after the view class is identified. For rapid view class identification, relational summaries were derived from the relational pyramids. If the relational pyramid has a relation R with c tuples given by

$$\{[(N_1, t_{1,j}), ..., (N_n, t_{n,j})] \mid j = 1, ..., c\}$$

Here, $t_{i,j}$ is a tuple from a lower level of the pyramid and N_i is the name of the relation that tuple $t_{i,j}$ comes from.

The summary has a corresponding relation P_L with a single tuple $[(N_1, ..., N_n), c]$ representing those c tuples. For example, if the collinear relation has four types of the form [ROCK, r), (BRAVO, b)], then the collinear summary relation has one tuple [(ROCK, BRAVO), 4], indicating that there are four collinearity relationships between a rock junction and bravo junction in line drawing of this view class.

The online system keeps an evidence accumulator for each view class, initialized to zero. For exact matching, the system traverses the summary structure, it adds one to the accumulators of all the view classes on the list attached to that tuple in the index. The view class or classes with maximal evidence are selected. For inexact matching, when considering summary tuple $[(N_1, ..., N_n), c]$, the system adds $e^{-k^2/2}$ to the accumulators of those view classes on list attached to $[(N_1, ..., N_n), c + k]$ and $[(N_1, ..., N_n), c - k]$ for $k = 0, ..., K$, where K is the maximum amount of deviation allowed.

(ii) Pose Determination with View Class – Once the view class has been determined, feature correspondences must be found the determine the pose. There are two possible approaches. First is to use a general purpose matching procedure for each view class, with the particular relationships and

constraints of that view class used to prune the search for a solution. The relational pyramid structure is a hierarchical, relational structure that can constrain the feature matching at each possible level of the pyramid. A strategy that tries to use higher level features first and propagate the results to lower levels of pyramid can result in a fast match. The second possibility is to develop a customized procedure for each view class. A preprocessing program analyzes each view class, possibly using training data or theoretical analysis to determine reliable features, and then selects a sequence of features to look for in the matching.

Q.11. Write brief note on model database organization.

Ans. An important problem for robot vision system is to organize the database of models in a way that system allows rapid access to the most likely candidate models, several approaches to this problem are available. For example, the original affine-invariant matching technique stores (model, basis) pairs for all possible models in its hash table structure. If there are x models, each with an average of y interest points, then total number of such pairs stored in the database is given by

$$= (y)\frac{\lfloor x}{\lfloor x-4}$$

which can grow quite large. If the object models can be characterized by their global attributes, then, like the view-class determination approach, the object classes can be chosen by some type of decision tree or other statistical or syntactic classifier. Such classification may be very simple, but very useful.

A scheme for organizing relational models based on the relational-distance metric was developed in 1982 by Sharpiro and Haralick. The idea of this approach, is to group similar relational models into clusters and to select a representative for each cluster. An unknown object would be compared with the cluster representatives and then with the models in those clusters deemed similar enough. Clustering can be done by any clustering algorithm that can work with distance between object, i.e., as opposed to points in n-dimensional space. The representative should be a relational description within the cluster that somehow best represents that cluster.

Assume that a cluster A has been constructed. For any relational description D_x in A, define the total distance of D_x with respect to A is given by,

$$T(D_x, A) = \sum_{D \in A} R_D(D_x, D).$$

The relational description D_y that satisfies $T(D_y, A) = \min_{D \in A} T(D, A)$ is used as the representative of the clusters. If the clusters are large, this entire process can be repeated to create a hierarchical structure of clusters and representatives.

☯ ☯

UNIT 4

FACET MODEL RECOGNITION – LABELING LINES, UNDERSTANDING LINE DRAWINGS, CLASSIFICATION OF SHAPES BY LABELING OF EDGES, RECOGNITION OF SHAPES, CONSISTING LABELING PROBLEM, BACK-TRACKING ALGORITHM

Q.1. Discuss about the facet model in details.

Ans. The facet model is a powerful tool in image processing. Its uses range from edge detection, background normalization, shape and surface topography, to image segmentation procedures involving detection of corners, curves, valleys, and ridges. The facet model principle is based on the minimization of the error between the image thought of as a piecewise continuous gray level intensity surface and the observed data from the physical scene. The image is considered as a noisy discretized sampling of the surface. The general forms of the facet model include piecewise constant, piecewise linear, piecewise quadratic, and piecewise cubic. In the constant model, each region in the image has a constant gray level. In the sloped model, each region has a gray level surface that is a sloped plane. The model used in this work is the cubic polynominal defined by equation (i),

$$f(x, y) = k_1 + k_2x + k_3y + k_4x^2 + k_5xy + k_6y^2 + k_7x^3 + k_8x^2y + k_9xy^2 + k_{10}y^3 \quad ...(i)$$

where f(x, y) is the gray level value at pixel location (x, y) whose neighbourhood is to be fitted. A local vector of the ten coefficients, computed as weighted sums of the values in the local neighbourhood, is found for each pixel (x, y). A discrete orthogonal polynomial basis permits independent estimation of each coefficient as a linear combination of the data values in the neighbourhood of (x, y). Those polynomials are given by equation (ii) for the 1-D case. The 2-D polynomials are obtained by taking the tensor product of the 2 sets of 1-D polynomials.

Let the discrete integer index set R be symmetric in the sense that $r \in R$ implies $-r \in R$. Let $P_n(r)$ be the n^{th} order polynomial. The discrete polynomials are iteratively constructed as follows –

Define $P_0(r) = 1$. Suppose $P_0(r), ..., P_{n-1}(r)$ have been defined.

$$P_n(r) = r^n + a_{n-1}r^{n-1} + ... + a_1 r + a_0 \qquad ...(ii)$$

$P_n(r)$ must be orthogonal to each polynomial $P_0(r), ..., P_{n-1}(r)$. We then have the set of n linear equations

$$\sum_{r \in R} P_k(r)\left(r^n + a_{n-1}r^{n-1} + ... + a_1 r + a_0\right) = 0,\ k = 0,....,n-1 \qquad ...(iii)$$

Solving for the set of equations yields the set of discrete orthogonal polynomials

$$P_{i+1}(r) = rP_i(r) - \beta_i P_{i-1}(r) \qquad ...(iv)$$

where $$\beta_i = \frac{\sum rP_i(r)P_{i-1}(r)}{\sum_{r \in R} P_{i-1}(r^2)} \qquad ...(v)$$

$P_0(r) = 1,\ P_1(r) = r$

The first five polynomials are given as

$$\left.\begin{aligned} P_0(r) &= 1 \\ P_1(r) &= r \\ P_2(r) &= r^2 - \frac{\mu_2}{\mu_0} \\ P_3(r) &= r^3 - \left(\frac{\mu_4}{\mu_2} . r\right) \\ P_4(r) &= r^4 + \frac{(\mu_2\mu_4 - \mu_0\mu_6)r^2 + \mu_2\mu_6 - \mu_4^2}{\mu_0\mu_4 - \mu_2^2} \end{aligned}\right\} \qquad ...(vi)$$

where $\mu_k = \sum_{s \in R} s^k$.

The facet model consists of solving an equal weighted least square fitting problem by minimizing the error

$$e^2 = \sum_{r \in R}\left[d(r) - \sum_{n=0}^{K} a_n P_n(r)\right]^2 \qquad ...(vii)$$

in terms of the a_n coefficients. d(r) is the data value observed (gray level values). The coefficients of the bivariate cubic of equation (i), $k_1, k_2, ..., k_n$ can then be determined. An error image describing the quality of fit is also generated. Given, the ten coefficients k_i defining the polynomial at pixel location (x, y), a number of topographic measurements can be determined. Image intensity surface patches are labeled and grouped according to the categories defined by monotonic, gray level, and invariant functions of directional derivatives, namely the gradient and the Hessian of the facets given by equation (viii).

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \text{ and } \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \qquad ...(viii)$$

The signs of those quantities are used to identify the region's label.

Q.2. Write short note on labeling lines.

Ans. In fig. 4.1, a cube resting on the floor, lines labeled with a + are caused by a convex edge, those labeled with a – are caused by a concave edge, and those labeled with a > are caused by matter occluding a surface behind it. The occluding matter is to the right of the line looking in the direction of the >, the

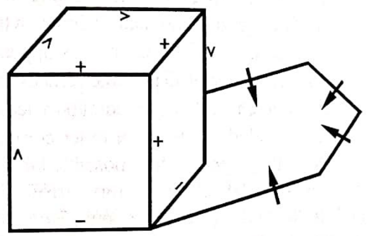


Fig. 4.1

Table 4.1 Vertex Catalogue

Visible Surfaces / Octants Filed	3	2	1	0
1				—
3				—
5			—	—
7		—	—	—
Occlusion				

occluded surface is to the left. If the cube were floating, one would label the lowest lines with < instead of with – . A systematic investigation can find the types of lines possibly seen around a trihedral corner, such corners can be classified by how many octants of space are filled by matter around them (one for the corner of a cube, seven for the inside corner of a room etc.). By considering all possible trihedral corners as seen from all possible viewpoints, Huffman and Clowes found that without occlusion, just four vertex types and only a few of the possible labelings of lines meeting at a vertex can occur. Fig. 4.2 shows views of one- and three-octant corners which give rise to all possible vertices for these corner types. The vertices appear in the first two rows of table 4.1, which is a catalogue of all possible vertices, including those arising from occlusion, in this restricted world of trihedral polyhedra. It is easy to imagine extending the catalog to include vertices for other corner types.

Note that there are four possible labels for each line (+ – > <), and thus 4^3 = 64 possible labels for the fork, arrow, and T and 16 possible labels for the cell. In the catalog, however, only 3/64, 3/64, 4/64, and 6/16, respectively, of the possible labels actually occur. Thus only a small fraction of possible labels can occur in a scene.

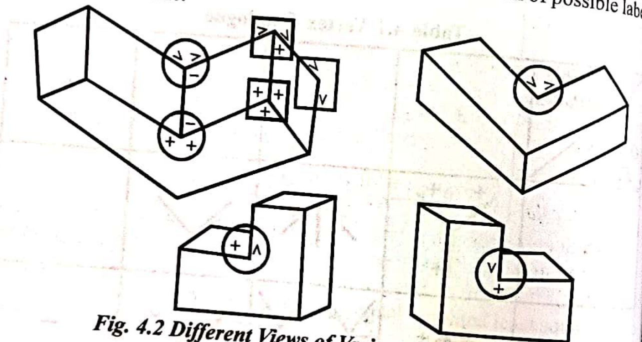


Fig. 4.2 Different Views of Various Corner Types

Q.3. What do you understand by line drawings ?

Ans. Line drawings have been the main medium of communication between human being about quantitative aspects of three dimensional object.

Line drawings are often ambiguous; interpreting them sometimes takes knowledge of everyday physics, and can require training. Such informed interpretation means that even drawings that are strictly nonsense can be understood and interpreted as they were meant. Missing lines in drawings of polyhedra are often so easy to supply as to pass unnoticed, or be "automatically supplied" by our model-driven perception.

Generalizing the line drawing to three dimensions as a list of lines or points is not enough to make an unambiguous representation, as shown by

fig 4.3, which illustrates that a set of vertices or edges can define many different solids. A line drawing nevertheless does convey three-dimensional information. For any set of N projection specifications, a wire-frame object may be constructed that is ambiguous given the N projections. However, for a given object, there is a maximum number of projections that can determine the object unambiguously.

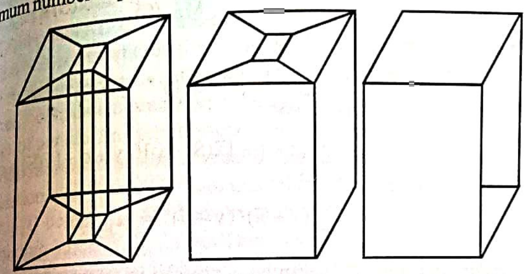


Fig. 4.3 An Ambiguous Representation

Line drawings were a natural early target for computer vision for the following reasons –

(i) They are related closely to surface features of polyhedral scenes.

(ii) They may be represented exactly; the noise and incomplete visual processing that may have affected the "line drawing extraction" can be modelled at will or completely eliminated.

(iii) They present an interpretation problem that is significant but seems approachable.

Q.4. Explain the shape recognition by moments.

Ans. Suppose f be a binary image function and suppose S = {(x, y) | f(x, y) = 1} represent a 2-D shape. For each pair of nonnegative integers (j, k), the digital (j, k)th moment of S is given by

$$M_{jk}(S) = \sum_{(x,y)\in S} x^j, y^k$$

$M_{00}(S)$ is then just #S. Moment invariants are function of the digital moments that are invariant under certain shape transformation. For 2-D shape recognition, we would like to have quantities that are moment invariants under translation, rotation, scaling and some kind of skewing.

The COG (center of gravity) $(\bar{x}, \bar{y})$ of S can be expressed in terms of some moments are defined as –

$$\bar{x} = \frac{M_{10}(S)}{M_{00}(S)} \text{ and } \bar{y} = \frac{M_{01}(S)}{M_{00}(S)}$$

Using the COG, we can define the central (j, k)th moment of S by

$$\mu_{jk} = \sum_{(x,y)\in S} (x-\bar{x})^j (y-\bar{y})^k$$

The central moments are translation invariant, since if

$$S^* = \{(x^*, y^*) \mid x^* = x + p, y^* = y + q, (x, y) \in S\}$$

Then
$$\bar{x}(S^*) = \frac{\sum_{(x^*,y^*)\in S^*} x^*}{M_{00}(S)} = \frac{\sum_{(x,y)\in S}(x+p)}{M_{00}(S)} = \bar{x}(S) + p$$

Similarly,
$$\bar{y}(S^*) = \bar{y}(S) + q$$

and
$$\mu_{jk}(S^*) = \sum_{(x^*,y^*)\in S^*}[x^* - \bar{x}(S^*)]^j[y^* - \bar{y}(S^*)]^k$$

$$= \sum_{(x,y)\in S}\{x + p - [\bar{x}(S) + p]^j(y + q - [\bar{y}(S) + q]\}^k$$

$$= \sum_{(x,y)\in S}(x - \bar{x})^j(y - \bar{y})^k = \mu_{jk}(S)$$

The standard deviation (SD) can be expressed in terms of moments -

$$\sigma_x = \left[\frac{\mu_{20}}{M_{00}}\right]^{1/2}, \quad \sigma_y = \left[\frac{\mu_{02}}{M_{00}}\right]^{1/2}$$

Alt normalized the coordinates by their respective SD to obtain the normalized coordinates

$$x' = \frac{(x - \bar{x})}{\sigma_x}, \quad y' = \frac{(y - \bar{y})}{\sigma_y}$$

Thus the mean values of x' and y' are both 0, and variance are both 1. Normalizing by area, the normalized moments defined by

$$m_{jk} = \frac{\sum (x')^j (y')^k}{M_{00}}$$

These moments are invariant under translation, scale and in general, affine transformations of the form $x^* = px + q, y^* = ry + t$ (referred to as "stretching" and "squeezing" transformation), since if

$$S^* = \{(x^*, y^*) \mid x^* = px + q, y^* = ry + t, (x, y) \in S\},$$

then we have

$$m_{jk}(S^*) = \frac{\sum_{(x^*,y^*)\in S^*}\left(\frac{x^* - \bar{x}(S^*)}{\sigma_x(S^*)}\right)^j\left(\frac{y^* - \bar{y}(S^*)}{\sigma_y(S^*)}\right)^k}{M_{00}(S^*)}$$

$$= \frac{\sum_{(x,y)\in S}\frac{p^j[x - \bar{x}(S)]^j}{p^j(\sigma_x)^j(S)}\frac{r^k[y - \bar{y}(S)]^k}{r^k(\sigma_y)^k(S)}}{M_{00}(S)} = M_{jk}(S)$$

Q.5. Discuss about the consisting labeling problem.

Ans. A consisting-labeling problem (CLP) arise in computer vision, in AI (artificial intelligence), and in science and engineering in general. An N-ary CLP is a 4-tuple component, namely, U, L, T and R. The first component U of the 4-tuple is a set of M units U = {1, ..., M} which are the objects to be labeled. The second component L is the set of possible labels. The third component T is called the unit-constraint relation. T is an N-ary relation over the set U of units. If an N-tuple $(u_1, u_2, ..., u_N)$ belongs to T, then we say that units u_1, u_2 ....., u_N mutually constrain one another. Groups of regions in an image that are adjacent or groups of parallel line segment can potentially mutually constrain one another. Finally, the fourth component R is called the unit-label constraint relation. R is an N-ary relation over the set U × L of unit-label pairs. If an N-tuple $[(u_1, l_1), (u_2, l_2), ..., (u_N, l_N)]$ belongs to R, then the units $u_1, u_2, ..., u_N$ may be assigned the corresponding labels $l_1, l_2, ..., l_N$. Thus the elements of R are allowable labeling of ordered size N subsets of unit set U. The only groups of units that are constrained are the N-tuples of T, the unit constraint relation. A labeling of subset $\hat{U} = \{u_1, u_2, ..., u_N\}$ of U is a mapping $f : \hat{U} \to L$ from $\hat{U}$ to L. A labeling f of a subset $\hat{U}$ of the units is consistent if whenever u_1, u_2 ..... and u_N are in $\hat{U}$ and the N-tuple $(u_1, u_2, ..., u_N)$ is in T, then $[(u_1, f(u_1)), (u_2, f(u_2)), ..., (u_N, f(u_N))]$ is in R. The goal of the consistent labeling problem is to find one or all consistent-labelings of the unit set U.

Q.6. *Explain the following –*

(i) The N-Queens problem (ii) The Latin-square puzzle.

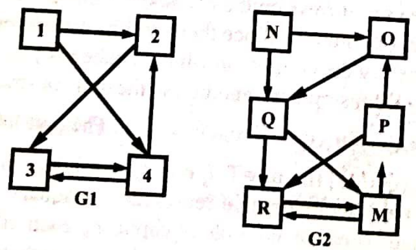
Ans. *(i) The N-Queens Problem* – This problem can be formulated as consistent-labeling problem. In this problem, we are given N × N chessboard and N queens. The queens must be placed on the chessboard in such a way that no queen can capture any other queen. This means that no two may be in the same row, same column, or same diagonal of the chessboard. The N-queens problem can be modeled as a consistent-labeling problem in which the unit set U = {1, 2, 3, ..., N} is the set of rows on the chessboard and the label set L = {1, 2, 3, ..., N} is the set of columns. Since there will be exactly one queen per row, a labeling will specify the column on which a queen is placed in each row. Every pair of rows will constrain one another, so the unit constraint relation T will be the set $T = \{(u_i, u_j) \mid u_i, u_j \in U \text{ and } u_i \neq u_j\}$. The unit-label constraint relation $R = \{[(u_i, l_i), (u_j, l_j)] \mid (u_i, u_j) \in T, l_i \neq l_j, l_i, l_j \in L, |u_i - u_j| \neq |l_i - l_j|\}$ includes these pairs of unit-label pairs that represent two squares on the chessboard on which two queens can stand without capturing each other.

(ii) The Latin-square Puzzle – This problem is also an example of consistent labeling problems. The Latin-square puzzle is an N × N matrix with

N^2 objects that must be arranged on the matrix, one per square. We will consider a 4 × 4 puzzle for ease of illustration. In this case each object is one of four colours A = {pink, orange, blue, red} and has one of four shapes B = {rectangle, triangle, square, circle}. The problem is to arrange the objects such that each row, each column, and each of the two main diagonals of the matrix contains exactly one object of each color and exactly one object of each shape. One way of modeling the Latin-square puzzle is to let the 16 squares of the matrix be the set of unit U = {1, 2, ...,16}. Then the labels are the objects to be placed on the squares such as pink square, orange triangle, red rectangle, orange rectangle, and so on. We can represent L as the Cartesian product set L = A × B. The constraints are along the rows, columns, and main diagonals, so we can model T as a quaternary constraint T = {(u_1, u_2, u_3, u_4) | u_1, u_2, u_3 and u_4 all lie in the same row, column or diagonal}. The unit-label constraint relation R would then consist of quadruples of unit label pairs of the form {[u_1, (a_1, b_1)][u_2, (a_2, b_2) [u_3, (a_3, b_3)], [u_4, (a_4, b_4)]} where (u_1, u_2, u_3, u_4) is in T. $(a_i b_i) \in L$ for i = 1, ..., 4 and when $i \neq j$, $a_i \neq a_j$ and $b_i \neq b_j$.

Q.7. Explain the backtracking tree search. Also write its algorithm.

Ans. A backtracking tree search starts with the first unit of U. This unit can potentially match each label in set L. Each of these potential assignments is a node at level one of the tree. The algorithm selects one of these nodes, makes the assignment, selects the second unit of U, and begins to construct the children of the first node, which are nodes that map the second unit of U to each possible label of L. At this level some of the nodes may be ruled out because they violate the constraints. The process continues to level |U| of the tree. The path from the root node to any successful nodes at level |U| are the consistent labelings. A simple digraph matching problem as shown in fig. 4.4, a subgraph of graph G2, that is isomorphic to graph G1. The possible labels for each unit were chosen on the basis that indegree [f(u)] ≥ indegree (u) and outdegree[f(u)] ≥ outdegree(u) for all units u.



Unit	Possible Labels
1	M, N, O, P, Q
2	M, O, P, Q, R
3	M, O, P, Q, R
4	M, O, P, Q

Fig. 4.4

A portion of the backtracking tree search is shown in fig. 4.5.

Algorithm of Backtracking Tree – We denote, the set of unit U, the set of label L, the unit-constraint relation T, the unit label constraint relation R and the partial labeling accumulated by f.

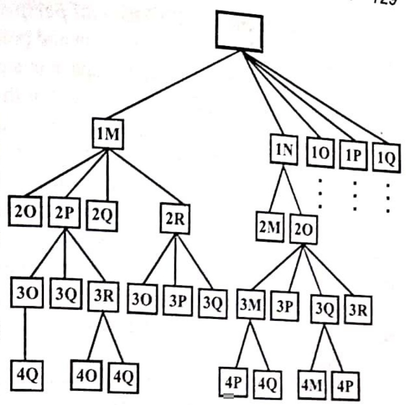


Fig. 4.5 Backtracking Tree Search

Procedure –

treesearch (U, L, f, T, R);
    u : first (U);
for each $l \in L$ **do**
    $f' = f \cup \{(u, l)\}$;
    OK : = true
for each N-tuple (u_1, ..., u_N) in T containing component u and whose other components are all in domain (f) **do**
if ((u_1, $f'(u_1)$),......,(u_N, $f'(u_N)$)) is not in R
    then begin OK : = false, break **end**;
        end for;
if *OK* **then**
    begin
        U′ = remainder(U);
        if isempty(U′)
        then output (f′)
        else treesearch (U′, L, f′, T, R);
    end
end for
end treesearch;

PERSPECTIVE PROJECTIVE GEOMETRY, INVERSE PERSPECTIVE PROJECTION, PHOTOGRAMMETRIC FROM 2D TO 3D

Q.8. Write short note on perspective projection geometry.

Ans. A pinhole camera is the simplest imaging device which, however captures accurately the geometry of perspective projection. Rays of light enters the camera through an infinitesimally small aperture. The intersection of the light rays with the image plane form the image of the object. Such a mapping from three dimensions onto two dimensions is called perspective projection.

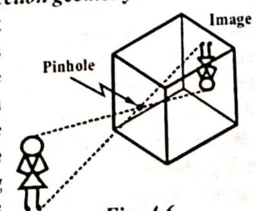


Fig. 4.6

Q.9. Explain the one-dimensional perspective projection.

Ans. The camera lens is at the origin and points directly down the y-axis, as shown in fig. 4.7. In order to keep the image in a positive orientation, we assume that the image line is at a distance d in front of the camera lens and that the lens projects forward to it. This eliminates the problems of left-right reversal in an image behind the lens. The image line for this example is parallel to the x-axis. According to the geometric ray optics model for the lens, the lens will focus a point (m, n) onto the image line, which is a line parallel to the x-axis and at a distance d directly in front of the lens. The distance d is known as the camera constant. The position on the line is determined by where the line from (m, n) to the origin intersects the image line. Hence the perspective projection has coordinates (md/n, d) in the original two dimensional coordinate system. Relative to the one-dimensional coordinate system of the image line, the coordinate is md/n. Note that, both the numerator and the denominator of md/n are linear combinations of m and n. If the numerator and denominator were computed by an appropriate linear transformation, the 1-D perspective coordinates could be computed by taking ratios of components of the transformed vector.

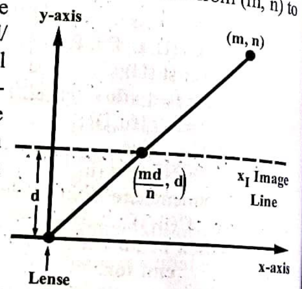


Fig. 4.7

This can be shown by using homogeneous coordinates. The point (m, n) can be represented as (m, n, 1) in the homogeneous coordinate system. The first linear transformation translates the point (m, n, 1) down the y-axis by a distance of d. The second transformation takes the perspective transformation to the image line. Hence, the 1-D image line coordinates, for the point are then given by $x_1 = a/b = md/n$.

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1/d & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m \\ n \\ 1 \end{bmatrix}$$

Q.10. Discuss about the three dimensional perspective projection.

Ans. The optic axis of a camera lens lies along a line parallel to the z-axis. To obtain the image frame coordinates for a given point in three dimension space, we first translate this point to a three dimension coordinate system centered at the lens of the camera. Then we translate along the z-axis by a distance d to the desired location of the projection image plane, and finally we take the perspective transformation. Perspective transformation is done by using a homogeneous coordinate system that assumes an arbitrary position of the lens. Let (x, y, z) be the original coordinates of a point in three dimension space. Let (x_0, y_0, z_0) be the position of the lens, called the center of perspectivity, and let (a, b) be the coordinates of the perspective projection of (x, y, z) on the image projection plane. Then $a = x^*/t^*$ and $b = y^*/t^*$, where

$$\begin{bmatrix} x^* \\ y^* \\ t^* \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}}_{\text{Perspective Projection}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Translation to Projection}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Translation to Lens}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0/d \end{bmatrix}$$

Thus

$$a = d\frac{x - x_0}{z - z_0} \quad \text{and} \quad b = d\frac{y - y_0}{z - z_0}$$

Q.11. Define the term inverse perspective projection.

Ans. Consider, a point whose perspective projection is (a, b) in the coordinate system of the image projection plane that is at a distance d in front of the camera lens has 3D coordinates (a, b, d). Since the camera lens is the origin, a ray passing through the point (a, b, d) and the origin consists of all multiples of (a, b, d). Furthermore, since the origin is the center of perspective projection (a, b). We call the line L,

$$L = \left\{ \begin{bmatrix} x \\ y \\ z \end{bmatrix} \middle| \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lambda \begin{bmatrix} a \\ b \\ d \end{bmatrix} \right\}$$

the inverse perspective projection of the point (a, b).

Q.12. Write short note on photogrammetric terminology.

Ans. Basic terminology used in photogrammetric is are as follows –

(i) Exterior Orientation – The exterior orientation of a camera is specified by all the parameters that determine the pose of the camera in the world reference frame. The parameters consist of the position of the center of perspectivity and the direction of the optical axis. Specification of the exterior orientation therefore requires three rotation angles and three translation parameters and it is accomplished by obtaining the 3D coordinates of some control points whose corresponding position on the image is known.

(ii) Interior Orientation – The integrior orientation of a camera is specified by all the parameters that determine the geometry of a bundle of three dimension rays from the measured image coordinates. The parameters

of interior orientation relate the geometry of ideal perspective projection to the physics of a camera. The parameters include the camera constant, the principal point and the specification of the lens distortion. Complete specification of the orientation of a camera is given by the interior and exterior orientations.

(iii) Relative Orientation – The relative orientation of one camera relative to another constitutes a stereo model and is specified by five parameters – three rotation angles and two translations. When two cameras are in relative orientation, each pair of corresponding rays from the two cameras intersect in three dimension space. The scale cannot be determined by relative orientation. The process of determining relative orientation assumes that the interior orientation of each camera is known.

(iv) Absolute Orientation – The absolute orientation involves the orientation of a stereo model in a world reference frame. This orientation requires the determination of seven parameters, such as the scale, the three translation parameters, and the three rotation parameters. It is accomplished by obtaining the three dimension coordinates of some central points whose position on the stereo image can be determined.

IMAGE MATCHING – INTENSITY MATCHING OF 1D SIGNALS, MATCHING OF 2D IMAGE, HIERARCHICAL IMAGE MATCHING

Q.13. Explain about the image matching.

Ans. Image matching is an essential and difficult task in digital photogrammetry and computer vision. It is the foundation of computer vision applications, such as camera calibration, three dimensional reconstruction, intelligent monitoring and motion analysis. Image matching is used for finding corresponding pixels in a pair of image which allows 3D reconstruction by triangulation. Image matching is relatively easy when encountered with good image texture conditions. However, on relatively poor textural image, image matching is a difficult and challenging problem. Most of the traditional digital photogrammetry systems require lots of human interactions to remove the errors in the matching results when dealing with poor textural images.

A lot of efforts have been devoted in the field of photogrammetry and computer vision to improve the reliability, automation, and efficiency of image matching which can be generally divided into two classes based on the matching primitives. One is area-based matching and the other is feature-based matching.

(i) Area-based Matching – Area-based matching usually works directly on local image windows, and it can acquire dense correspondences. It uses the grey value of the whole image to measure the similarity of two images directly. And a certain method is employed to search the point where the similarity measurement is the biggest. There are many area-based matching methods such as maximization of mutual information, correlating method, conditional entropy method, joint entropy method and so on. Although area-based matching is the most widely used, there exists some shortcomings such as huge computation, long time of matching and sensitivity to rotating, scaling and distort.

(ii) Feature-based Matching – The common image feature includes point feature, straight line, edge, shape, closed area, statistical moment, etc. By far, feature extraction algorithm can be divided into three main classes – one is point feature extraction operator such as Förstner operator, Harris operator and Susan operator, the second is linear feature extraction operator (such as Canny operator, LoG operator), and the third is surface extraction operator mainly through region segmentation. Generally speaking, feature-based matching has the advantage of being simple to operate, rapid matching speed and high precise matching rate, but it also requires human intervention and the obtaining of feature points is a bit difficult. Besides, it is only suitable for simple images with significant geometric features.

Q.14. Discuss about the intensity based matching techniques in 1-D signal.

Ans. Intensity based matching techniques directly refer to the model equation

$$g'(r',c') = T_I\{g''[T_G(r'',c'';p_G)];p_I\} \quad ...(i)$$

and aim at estimating and evaluating the parameters p_G and p_I. To give insight into the principles, we first develop methods for matching 1D signals. As even this task is demanding when taking the statistical properties of the data into account, we restrict this discussion to the case in which one of the signals is assumed to be perfectly known. This situation is relevant to an object location procedure. The model then can be written as

$$g(x) = T_I\{f[T_G(y;p_g)];p_I\} + n(x) \quad ...(ii)$$

Here, g', g'', r' and r'' have been replaced by g, f, x and y respectively, and the observational noise component n(x) is stated explicitly. Let f(y) is given by sampled data and a fitting or interpolation scheme that allows one to use a derived or estimated continuous f(y) from which the first derivatives can be obtained analytically. This is just like the facet model. Because of the highly nonlinear character of the estimation problem, we always, assume that approximate values $p_I^{(0)}$ and $p_G^{(0)}$ are known from some prior information or prediction scheme. This initial approximation permits us to replace the nonlinear problem by a linear substitute problem whose solution then gives rise to better approximations. In this way we naturally arrive at an iterative solution to the nonlinear problem. We therefore always assume that second-order effects are

sufficiently small specifically, (i) when interpolating f or its derivate, (ii) when neglecting the random nature of f if it is derived from real data, or (iii) when deriving variance for the estimated parameters.

Q.15. *Write short note on epipolar geometry.*

Ans. Image matching can be tremendously simplified if the relative orientation known as the two-diamensional search space is reduced to a one dimensional one by the so called epipolar geometry inherent in the oriented image pair. The general setup of cameras is shown in fig. 4.8. The projection centers A' and A" form the baseline of length b, the principal points B' and B" are assumed to be the origin of the two image coordinate systems (u',v') and (u",v") derived from the pixel coordinates (r',c') and (r",c") by using the interior orientation. The object point P(x, y, z) then is mapped into P'(u',v') and P"(u",v") in the image planes μ' and μ". Because of the geometric model of the perspective projection – specifically the collinearity condition – the five points P, A', A", P' and p" lie in one plane, the so called epipolar plane ∈ (P) associated with P. The intersection lines of ∈ (P) with μ' and μ" result in the two epipolar lines e'(P) and e"(P) associated with P. For points P_1 not sitting in the same epipolar plane, we obtain different pairs of epipolar lines. All epipolar planes form a pencil of planes passing through the baseline b = (A'A"). The epipolar lines intersect in the epiploles F' and F", which are the intersection of the baseline b intersects with the image planes μ' and μ", respectively Thus in general epipolar lines are not parallel.

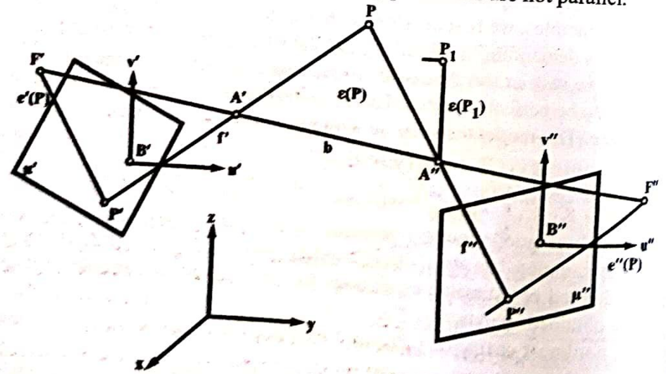


Fig. 4.8

Q.16. *Explain the hierarchical image matching.*

Ans. The hierarchical image matching method first uses a SIFT algorithm and RANSAC approach to obtain a few reliable correspondences, and construct

an initial triangulation. The SIFT algorithm is proved to be able to produce robust but relative sparse corresponding points invariant to moderate scale changes or distortions, which is ideal for the purpose of generating a certain number of well distributed matching points for the initial triangulation. In the SIFT descriptor, each interest point is characterized by a vector with 128 unsigned eight-bit numbers generated from a local region, which defines the multi-scale gradient orientation histogram. The matching is performed by measuring the similarity between the two vectors associated with the two matching points.

The RANSAC approach is used to detect and eliminate possible blunders from the previous SIFT matching results. It starts by randomly selecting a subset of the matched corresponding points. From the chosen matched points, a fundamental matrix can be calculated based on which a model is then built. This model is evaluated by determining whether each pair of corresponding points fit reasonably well to it. This is used as a criterion to determine the best model which has the largest number of correct corresponding points. This process is repeated to find the best model. Those matched points which do not fit for the final best model are considered as blunders and eliminated from the initial matching point set.

After the seed points are extracted at the top level, an initial triangulation can be constructed and an area-based image matching with feature points and grid points is conducted at the top level again.

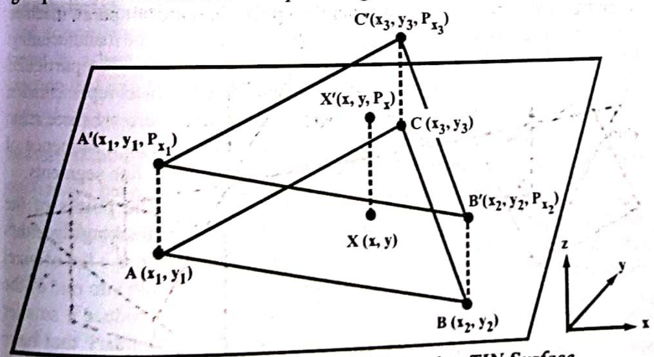


Fig. 4.9 Interpolation of x Parallax using TIN Surface

In the process of hierarchical strategy, image matching is first conducted on the lowest resolution. The matched points are then transferred to the next level (of higher resolution) where additional feature points could be matched. This process repeats until it reaches up to the original image level. At a subsequent level, points from upper level are matched again to achieve higher precision. A TIN (Triangulated Irregular Network) surface of parallaxes is

generated from these matched points using the Delaunay triangulation. This TIN is used to estimate the correspondence of additional feature points.

As shown in fig. 4.9, a point $X(x, y)$ is inside of a Delaunay triangle formed by points $A(x_1, y_1), B(x_2, y_2), C(x_3, y_3)$, the x parallax is interpolated based on the TIN surface formed by $A'(x_1, y_1, p_{x_1}), B'(x_2, y_2, p_{x_2}), C'(x_3, y_3, p_{x_3})$. The x parallax of point $X(p_x)$ is interpolated as the x coordinate of point X', which is the intersection between the 3-D plane $A'B'C'$ and the line XX' parallel to z-axis. Parallax in y direction (p_x) can be calculated using the same strategy, only using y parallax as z coordinates. Finally, the estimated corresponding point coordinate (x', y') is defined as –

$$x' = x + p_x$$
$$y' = y + p_y$$

OBJECT MODELS AND MATCHING – 2D REPRESENTATION, GLOBAL VS LOCAL FEATURES

Q.17. Discuss about the two dimensional representation.

Ans. 2D shape analysis is useful in a number of applications of machine vision, including medical image analysis, aerial image analysis and manufacturing. The method used for shape recognition often depends on the particular representation selected. The most common example of 2D object representation is ***boundary representation.*** In boundary representation, there are three main ways to represent the boundary of an object such as first, as a sequence of points, second, by its chain code and third, as a sequence of line segments.

(i) A Sequence of Points Representation – The points of the boundary come from some kind of border-following or edge-tracking algorithm performed on a digital image. The result of such an operation is a list of pixel coordinates. The list can be maintained as a whole, converted into one of the other two main boundary representation, or processed to produce a smaller list of interest points. Interest points are points on the boundary that have some special property that makes them useful in a given matching algorithm.

One method of extracting these interest points from the original sequence of boundary points of the curve is the curve-partitioning algorithm. Given a point A on the curve and a fixed arc length *l*, there is a set of chords that have arc length *l* and span the part of the curve containing A. Let d(A, C) be the perpendicular distance from a point A to a chord C whose span consi[illegible] A, and let M (A, C) be the maximum distance from A to all such chords. [illegible]ion

point of the curve if the value of M(A, C) is a local maximum and also exceeds a threshold t (*l*). This method finds points of high curvature along the boundary. It can be modified to select a point A that is the median point in a sequence of points $<A_1, \ldots, A_n>$ for which $M(A_i, C)$, i = 1, ..., n, are all local maxima. In this way it detects not only very sharp corners but also points of high curvature along the boundary that are part of a section of approximately constant curvature.

(ii) The Chain-code Representation – Refer to Q.2, Unit-2.

(iii) A Sequence of Line Segments Representation – This is a third common representation for the boundary of a 2D shape. This representation is generally used after the original sequence of boundary points has been segmented into a set of line segments representing near-linear portions of the boundary. Once the sequence of line segments has been computed by some method, it can be converted into a model of the shape that can be used in shape recognition or other matching tasks. A model for representing the matching sequences of line segments was introduced by Davis. According to Davis, a line segment sequence by the sequence of junction points $\langle X_i, Y_i, \alpha_i \rangle$ where a pair of lines meet at coordinate location (X_i, Y_i) with angle magnitude α_i. Given a sequence $A = A_1, A_2, \ldots, A_n$ of junction points representing the boundary of a model object A and a similar sequence $B = B_1, B_2 \ldots B_m$ representing the boundary of a test object B. The goal is to find an association $E = \{1, 2, \ldots, m\} \Rightarrow \{1, 2, \ldots, n\} \cup \{\text{missing}\}$ that satisfies $i < j \Rightarrow E(i) < E(j)$ or either E(i) = missing or E(j) = missing. Davis used constraints on both sides (line segments) and angles to define what is meant by a best mapping for this problem. Let M(i, j) be a local evaluation function that measures the goodness of the match of junction i of A to junction j of A, based on the difference between the angles α_i and α_j. Let $S_{ij}(i', j')$ be a measure of the consistency of mapping junction i to junction i′ and junction j to junction j′, based on the difference between the segment lengths of B_iB_j and A_iA_j. The cost of a mapping E is given by –

$$C(E) = \sum_{i=1}^{m} M[i, E(i)] + \sum_{i=1}^{m}\sum_{j=1}^{m} S_{ij}[E(i), E(j)] + P(m_B) + P(m_A)$$

Here, P is a penalty function for missing angles.

Q.18. Explain the following terms –

(i) Global feature (ii) Local feature.

Ans. (i) Global Feature – A 2D object can be thought of as a binary image. The pixels of the object have value 1, and the pixels outside the object have value 0. Because of this relationship, it is natural to represent by using some of the same features we used to represent binary images. Commonly

used features for 2D shape representation include area, perimeter, moments, circularity, and elongation.

(ii) Local Feature – A 2D object can also be characterized by its local features, their attributes, and their interrelationships. The most commonly used local features in industrial part recognition are holes and corners. Holes can be detected by a connected component procedure followed by boundary tracing or, if the shapes of the holes are known in advance, through the operations of binary mathematical morphology. Local features must be organized into some type of structure for matching. The most common type of structure is a graph whose nodes represent local features and their properties or measurement and whose edges represent relationships among the features.

UNIT 5

KNOWLEDGE BASED VISION – KNOWLEDGE REPRESENTATION, CONTROL STRATEGIES, INFORMATION INTEGRATION

Q.1. Explain the knowledge-based computer vision with suitable diagram.

Ans. Fig. 5.1 shows the typical architecture of a Knowledge-based Computer Vision (KBCV) system structured in two different sub-systems – low-level and high-level processing. At low-level processing, the raw-data is processed by signal processing algorithms, usually generating feature descriptions, sets of symbolic descriptors which summarizes characteristics of data in a quantitative way. Examples of low-level processing algorithms are segmentation, color detection, texture detection, noise reduction, occlusion detection and so forth. These algorithms were the main focus of research in computer vision for a long time and had come a long way in terms of performance.

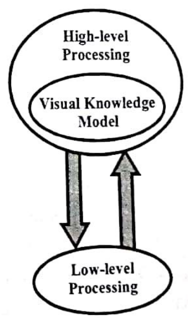


Fig. 5.1 Typical Architecture of KBCV

On the other hand, high-level processing is related to the interpretation and reasoning with visual data. It is usually built on top of the low-level processing algorithms, taking features descriptors as input and generating abstract, qualitative descriptions about the content of the visual data. This is called content descriptions. Ideally, the high-level processing can also act on the low-level processing adjusting its parameters to improved their performance based on generated content descriptions, creating a kind of feedback loop between the two levels (usually called bottom-up and top-down reasoning). High-level processing has been implemented using the various forms of

symbolic artificial intelligence, from rule-based to probabilistic systems. They usually employ some sort of a priori knowledge about the visual interpretation to be made.

Q.2. Explain in detail about knowledge representation.

Ans. A knowledge representation is a set of syntactic and semantic conventions to describe a piece of knowledge. In order to build knowledge bases the experts describe the contents of their knowledge using knowledge representation schemes. A knowledge representation scheme must allow to explicit what is important and must be easy to handle. Two main knowledge representation schemes are presented in the following, production rules and frames.

(i) Production Rules – Production rules are directly inspired from predicate logic and the inference capabilities of modus ponens. A production rule very generally expresses an "if-then" relationship. General syntax for production rules are as follows –

Rule ruleX
  comment : "a comment explaining the rule"
  if
    condition
  and/or
    condition
  then
    conclusion

Which means that if a set of conditions are true then the conclusion holds. For instance in the domain of the bridge game, if we want to express that during bidding one can open with a specific distribution. For example, we can use the production rule R1bridge –

Rule R1bridge
  comment : "bidding rule for opening"
  if
    balanced hand
  and
    12 to 15 head points
  then
    opening 1 no trump

This rule means that if we have a balanced hand and between 12 and 15 head points then we can open at the level of 1 no trump.

Each piece of knowledge can be expressed with a different rule. A knowledge base can contain hundreds of production rules. It is very easy to add, modify or remove a production rule in a knowledge base. This allows to extend or update a knowledge base. Production rules can be grouped in different sets to structure the knowledge and help the reasoning.

The power of expression of production rules relies on the kind of logic they are based. For rules based on propositional logic symbols represent all propositions or facts, everything is a constant. It was the case in the previous example for bidding rule R1bridge. A usual kind of production rules are rules based on 0+ order logic. The formalisms (object, attribute, value) or (attribute, value) are used; the values can be referenced. The rule R2 shows an example of 0+ order production rule, where the value of the temperature in the room is not directly given in the rule but referenced through the attribute room-temperature.

Example of 0+ order production rules are given below –

Rule R2
  comment : "use the heater if the temperature is too low"
  if
    room-temperature < 19
  then
    change heating-status to on.

For rules based on first-order logic or predicate logic quantifiers and variables can be used. The rule R3 shows an example of first-order production rule, where two variables M and N and one quantifier, $\exists$, are used.

Example of first-order production rules are given as –

Rule R3
  comment : "track mobile regions which are possible human beings"
  if
    $\exists$ mobile object M
  and
    shape of M = human
  and
    1.4 metres < size of M < 2 metres
  then
    track M

In conclusion we can say that production rules are simple, in particular in the case of propositional logic; they are very modular and readable. They allow fast modifications of a knowledge base and explanations are easy to provide to a user. On the contrary they have different drawbacks – the knowledge is fragmented, the uniform formalism for expressing the knowledge leads to a lack of efficiency for problem solving.

(ii) Frames – A frame is a knowledge representation scheme which comes from cognitive research on human reasoning. The hypothesis is that human beings refer when reasoning to prototypes already stored in memory and compare them to objects or events corresponding to new situations.

The concept of frames is a declarative knowledge representation scheme well adapted for structured object descriptions.

A frame is a set of attributes (or properties). Each attribute has several slots which describe the characteristics of the attribute. The attributes are very dependent on the nature of object it represents. The slots are predefined general characteristics useful for any kind of attribute; classical slots are the type, the current value, a default value, a possible range of values. Frame1 is an example of frame with n attributes and 4 kind of slots. A frame is useful to describe a general concept, as the concept of a chair.

Table 5.1 The General Frame Frame1

Attributes	*Slots*	*Slot Values*
1	type	type-value
	value	attribute-value
	default	default-value
2	type	type-value
	value	value
	range-of-values	range-value
3	type	type-value
	value	attribute-value
	range-of-values	range-value
	default	default-value
....	....	
n	type	type-value
	value	attribute-value
	range-of-value	default-value
	default	default-value

Q.3. Describe various image control strategies.

Ans. Some control strategies in image processing are as follows –

(i) Bottom-up Control Strategy – This strategy is used for problem solving that is data driven. It employs no object models in its early stages and only uses general knowledge about the world being sensed. In a computer vision system using a bottom-up control strategy, the observed image data is interpreted and aggregated. The interpretations and aggregations are then successively manipulated and aggregated until a sufficiently high level description of the scene has been generated.

(ii) Top-down Control Strategy – This strategy is used for problem solving that is goal-directed or expectation directed. A form of solution is generated or hypothesized. Assuming the hypothesis is true and using the information in the knowledge data base, the inference mechanism then infers, if possible, some consistent set of values for the unknown variables or parameters. If a consistent set can be inferred, then the problem has been solved. If a consistent set cannot be inferred, then a new form of solution is generated or hypothesized. In a computer vision system using a top-down control structure, the number or types of objects being sensed in the image is usually highly constrained and knowledge about the objects, relationships between objects, and object parts are all known. The system hypothesizes that the image shows a particular set of objects, infers values for parameters, and then tests to verify that the hypothesis is consistent with the observed data.

(iii) Hierarchical Control Strategy – This strategy is used for problem solving in which the given problem is solved by dividing it up into a set of subproblems, each of which encapsulates an important or major aspect of the original problem. Then each subproblem is successively divided into more detailed subproblems. The refinement continues until the most refined subproblems can be solved directly.

(iv) Blackboard Control Strategy – This strategy is used for problem solving in which the various components of the inference mechanism communicate with one another through a common working data storage area called the blackboard. When the blackboard has sufficient data to permit one component of the inference mechanism to make a deduction, the inference mechanism goes to work and writes its results on the blackboard where it becomes available for the other components of the inference mechanism. In this manner the inferred constraints are successively propagated and the required search is made more limited.

Q.4. Which approach to the information integration problem has been used in computer vision.

Ans. The Bayesian approach to the information integration problem has been used in computer vision. The Bayesian approach to information integration introduced by Pearl in 1987. Pearl defines a Bayesian belief network as a directed acyclic graph whose nodes represent propositional variables and whose arcs represent causal relationships. Assume that the nodes of the graph are variables $A_1, A_2, \ldots, A_N$. Each propositional variable A_i has a finite set of possible values $\{a_i\}$. An arc (A_j, A_i) from node A_j to node A_i indicates that the value of A_j is a direct cause of the value of A_i. Assume A_i has I possible values, and A_j has J possible values. For each pair of values (a_j, a_i), the strength of the causality is the conditional probability $P(a_i \mid a_j)$. These probabilities can be thought of as forming a $J \times I$ matrix associated with the arc.

The support set of a node is the set of the node's predecessors. Assume that node A_i has support set S_i. Then $P(a_i|s_i)$ denotes the joint conditional

probability of $A_i = a_i$, given the set of value $\{s_i\}$ for the support variables. The distribution corresponding to the entire network is then

$$P(a_1, a_2, ...a_n) = \prod_{i=1}^{n} P(a_i \mid s_i)$$

The matrices of strengths on each of the causal links are static input to the network. As information propagates through the network, the belief value B(a) of each proposition A = a changes. The dynamically changing belief value B(a) denotes the probability P(a|e) of A = a, given all the evidence e received so far. The idea for a portion of a belief network to be used in image analysis is shown in fig. 5.2.

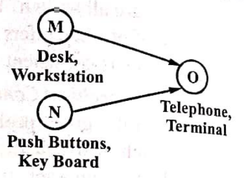


Fig. 5.2 Portion of a Belief Network

Where, nodes M, N and O represent objects to be identified. Each object has two possible labels – object M can be a workstation or a desk, object N can be set of pushbuttons or keyboard, and object O can be a telephone or a computer terminal. Labels of "desk" for M and "pushbuttons" for N tend to support the label of "telephone" for O, and labels of workstation for M and "keyboard" for N tend to support the label of "terminal" for O. The model of belief propagation defines how beliefs in the network are updated as new evidence enters the system. The idea for a propagation of beliefs through a piece of a belief network is shown in fig. 5.3.

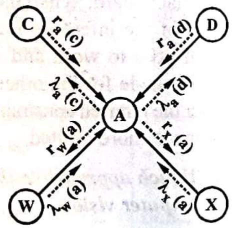


Fig. 5.3 Propagation of Beliefs through Piece of a Belief Network

Here the nodes C and D are predecessors of node A and nodes W and X are successors of node A.

Let P(a | c, d) be the fixed conditional probability matrix that relates the variable 'a' to its immediate parents c and d. Let $\pi_A(c)$ be the current strength of the causal support contributed by an incoming link to A. Let $\lambda_W(a)$ be the current strength of the diagnostic support contributed by an outgoing link from A. Causal support represents evidence propagating forward from parents to their children, whereas diagnostic support represents feedback from children to their parents. Updating a node A thus involves updating not only its belief function but also its λ and π functions. The formula of belief updating is

$$B(a) = \alpha\lambda_W(a)\lambda_X(a)\sum_{c,d} P(a \mid c, d)\pi_A(c)\pi_A(d)$$

where α is a normalizing constant that makes $\sum_a B(a) = 1$.

OBJECT RECOGNITION – HOUGH TRANSFORMS AND OTHER SIMPLE OBJECT RECOGNITION METHODS, SHAPE CORRESPONDENCE AND SHAPE MATCHING

Q.5. What is object recognition ? And its main components.

Ans. Object recognition is a technology in the field of computer vision. An object recognition system finds objects in the real world from an image of the world, using object models which are known as priori. This task is surprisingly difficult. Humans perform object recognition effort lessly and instantaneously. Algorithm description of this task for implementation on machines has been very difficult. The different steps in object recognition and introduce some techniques that have been used for object recognition in many applications. The architecture and main components of object recognition is shown in fig. 5.4.

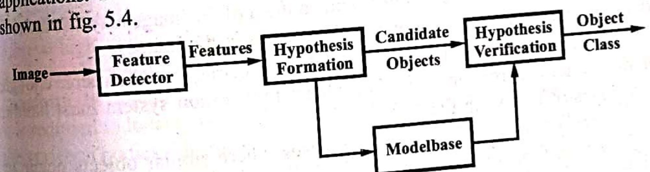


Fig. 5.4 Architecture of Object Recognition

The model database contains all the models known to the system. The information in the model database depends on the approach used for the recognition. It can vary from a qualitative or functional description to precise geometric surface information. In many cases, the models of objects are abstract feature vectors. A feature is some attribute of the object that is considered important in describing and recognizing the object in relation to other objects. Size, colour, and shape are some commonly used features.

The feature detector applies operators to images and identifies locations of features that help in forming object hypotheses. The features used by a system depend on the types of objects to be recognized and the organization of the model database. Using the detected features in the image, the hypothesizer assigns likelihoods to objects present in the scene. This step is used to reduce the search space for the recognizer using certain features. The model base is organized using some type of indexing scheme to facilitate elimination of unlikely

object candidates from possible consideration. The verifier then uses object models to verify the hypotheses and refines the likelihood of objects. The system then selects the object with the highest likelihood, based on all the evidence, as the correct object.

All object recognition systems use models either explicitly or implicitly and employ feature detectors based on these object models. The hypothesis formation and verification components vary in their importance in different approaches to object recognition. Some systems use only hypothesis formation and then select the object with highest likelihood as the correct object. Pattern classification approaches are a good example of this approach. Many artificial intelligence systems, on the other hand, rely little on the hypothesis formation and do more work in the verification phases. In fact, one of the classical approaches, template matching, bypasses the hypothesis formation stage entirely.

Q.6. What are the challenges faced in object recognition ?

Ans. The challenges faced in object recognition are as follows –

(i) Change in size, cropping out the background are some of the factors influencing the accuracy of the system. The accuracy of the model might change by scaling the image.

(ii) Adjusting brightness and contrast of the image may also make it difficult for the system to recognize the objects in the image.

(iii) There may be cases when the object might not be visible enough for the system to recognize it. The object recognition system must handle these cases of low visibility.

(iv) The system may fail in cases where similar objects occur in groups and are too small in size.

(v) Various lightning conditions and shadows in the image may also pose difficulty for the system to recognize the object.

Q.7. Give some applications of object recognition.

Ans. The applications of object recognition are as follows –

(i) Self-Driving Cars – Self driving cars may use object detection and recognition system to identify pedestrians and cars on the roads and then make the suitable decision in accordance.

(ii) Face Detection – Another application of object detection and recognition is face detection e.g. facebook recognizes people before they are tagged in images.

(iii) Medical Science – Object detection and recognition system may help medical science to detect diseases. For e.g. detecting tumors and various cancers.

(iv) Text Recognition – Text recognition deals with recognizing letters/symbols, individual words and series of words. Ex- recognizing handwriting of a person.

(v) Hand Gesture Recognition – Hand gesture recognition deals with recognition of hand poses, and sign languages.

Q.8. Explain Hough transform. Also write its advantages and disadvantages.

Ans. Paul Hough patented the technique of Hough transform in 1962 is a feature extraction method that can be used in image analysis and digital image processing. The aim of this technique is to produce a computer vision system that can detect arbitrary shapes within a sample image. The main purpose of this method is finding imperfect instances of objects within a certain class of shapes by a voting procedure. The classical Hough transform was mainly introduced for the identification of lines in images, but later the Hough transform has been modified and extended to identify the positions of arbitrary shapes within an image, most commonly the extended version indulged itself in finding circles or ellipses. In that case appropriate parametric representation is needed. Now-a-days there are a wide range of areas where the Hough transform can be implemented successfully such as in medical visualization or in order to achieve high accuracy in face recognition etc. The characteristics of Pupil and Iris under uncontrolled illumination can also be obtained by Hough circle transform. In objective spinal motion imaging assessment system (OSMIA), it is required to locate marker that can be used in determining the positions of the vertebral bodies. The measurement of vertebral motion has been a challenge to the field of biomechanics for many years but now several automatic approaches to these problems have been developed. The Hough transform has also been introduced in morphological image processing to detect nad estimate the number of red blood cells in the blood sample image.

The most common application of using the Hough transform is the linear transform for detecting straight lines in an image. In the image space, the straight line can be described as $y = mx + c$ where the parameter m represents slope of the line, and c represents the y-intercept. This form of representation is called the slope-intercept model of a straight line. The main idea behind using Hough transform is considering the features of the straight line not as discrete image points (x_1, y_1), (x_2, y_2), etc., but in terms of slope-intercept model. In general, the straight line $y = mx + c$ can be denoted as a point (m, c) in the parameter space where m represents the slope of the line and c represents the intercept.

Advantages and Disadvantages – The major advantage of Hough transform is that the pixels present on one line need not be adjacent to one another which can be exceptionally valuable when attempting to perceive lines with tiny breaks in them due to noise, or identification in partially occluded images.

With respect to hindrances of Hough transform, then it may give ambiguous results when objects are associated by chance. This plainly indicates another drawback that is the recognized lines are infinite lines defined by their (m, c) values, as opposed to finite lines with definite termination points.

Q.9. Discuss about the line Hough transform.

Ans. A suitable equation for describing a set of lines in a parametric form is –

$$x \cos\theta + y \sin\theta = r$$

where r is the length of the normal from the origin to this line and θ is the orientation of r with respect to the X-axis. A suitable way of describing this presentation is described in the fig. 5.5 which shows an object taken as an example and the fig. 5.6 shows the parametric representation of a line.

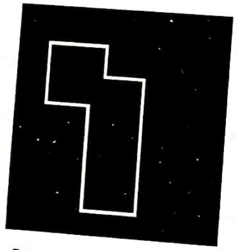


Fig. 5.5 Gradients of the Example Object

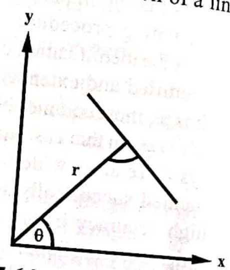


Fig. 5.6 Parametric Representation of a Line

When an image is analyzed the (x, y) parameters represents the well known pixels that are selected for analyzing. The pair of θ and r used for parametric representation is inserted into an accumulator.

Another approach to think about this fact is that all the lines that go through the point (x, y) are transformed into parametric space (θ, r) and then the relevant cell is increased by 1. Each and every single point present in the accumulator corresponds to certain line in the image. As this accumulator is discrete in nature it only consists of a set of all possible lines in R^2. It also corresponds to a set of sinusoid curves which intersects in some points.

A very useful feature of this algorithm is its robustness against noise and gaps present in the input image. This technique can be useful in real life applications such as in analyzing ultrasound images which contains some obvious noise that may causes problems in analyzing the features.

Q.10. Explain the circular Hough transform.

Ans. Feature extraction techniques, like, circle Hough transform (CHT) is used for detecting circles. It is a part of Hough transform. The reason for

this strategy is discovering circles in blemished image. In the Hough parameter space the circle candidates are created by "voting" and afterward nearby maxima is selected in a so-called accumulator matrix.

A change of a point in the x-y plane to the parameter space may be portrayed as the basic behind the general Hough transform. The parameter space is characterized by space of the object of interest.

The line is difficult to represent in parameter space, contrasted with the circle, transformation of the parameter of the circle to the parameter space can be done easily. The equation of the circle is –

$$r^2 = (x - m)^2 + (y - n)^2$$

where, m, n and r are three parameters of circle, where (m, n) is centre of the circle in the direction (x, y) respectively and r is its radius.

Circle can be represented in parametric form as –

$$x = m + r \cos\theta$$
$$y = n + r \sin\theta$$

Along these lines the circle's parameter space shall fit in with three dimensional, though the line just had a place with two dimensional. As the quantity of parameter expected to portray the shape increment and in addition the measurement of the parameter space expand, simultaneously the Hough transform complexity too increases. Therefore straightforward shape with parameter fitting in with two dimensional or at most three dimensional. The parametric representation of the circle, the range can be held constant or a fixed number of radii can be set.

At every edge-point we draw a circle with centre in the point with the craved range. The drawn circle attracts the parameter space, such that our x-axis, y-axis, and z-axis are m-value, n-value, and the radii respectively. Accumulator matrix has identical size as parameter space. Value in our accumulator matrix is incremented. Hence, we clear over vitality edge-point in the information image drawing circle with the wanted circle with preferred radii increasing and updating the value in our accumulator. Now, when each edge-point and each preferred radius is utilized, then we can get the number of circles passing through the individual coordinate stored at the accumulator. Hence the highest number relate to the centre of the circle in the image.

Q.11. Explain in brief about the patterns and pattern classes.

Ans. A pattern is an arrangement of descriptors. The name feature is used often in the pattern recognition literature to denote a descriptor. A pattern class is a family of patterns that share a set of common properties. Pattern classes are denoted $p_1, p_2, \ldots, p_N$, where N is the number of classes. Pattern recognition by machine involves techniques for assigning patterns to their respective classes automatically and with as little human intervention as

possible. The two principal pattern arrangements used in practice are vectors (for quantitative descriptions) and strings (for structural description). Pattern vectors are represented by bold lowercase letters like **x**, **y** and **z**, and have the n × 1 vector form.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

where each component x_i represents the i^{th} descriptor and n is the total number of such descriptors associated with the pattern. Sometimes it is necessary in computation to use row vector of dimension 1 × n, obtained simply by forming the transpose $\mathbf{x}^T$ of the preceding column vector.

The nature of the component of a pattern vector **x** depends on the approach used to describe the physical pattern itself.

Q.12. Discuss the recognition based on decision theoretic method.

Ans. The decision theoretic approaches to recognition are based on the use of decision function. Let $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ represent an n-dimensional pattern vector. For N pattern classes $p_1, p_2, ..., p_N$, the basic problem in decision theoretic recognition is to find N decision function $d_1(\mathbf{x}), d_2(\mathbf{x}), ..., d_N(\mathbf{x})$ with the property that, if a pattern **x** belongs to class p_i, then

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \qquad j = 1, 2, ..., N \quad j \neq i \qquad ...(i)$$

In other words, an unknown pattern **x** is said to belong to the i^{th} pattern class if, upon substitution of **x** into all decision functions $d_i(\mathbf{x})$ yields the largest numerical value. Ties are resolved arbitrarily.

The decision boundary separating class p_i from p_j is given by values of x for which $d_i(\mathbf{x}) = d_j(\mathbf{x})$ or, equivalently, by values of **x** for which

$$d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0 \qquad ...(ii)$$

Common practice is to identify the decision boundary between two classes by the single function $d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$. Thus $d_{ij}(\mathbf{x}) > 0$ for patterns of class p_i and $d_{ij}(\mathbf{x}) < 0$ for patterns of class p_j.

Q.13. Discuss about the structural methods.

Ans. Strings are the most practical approach in structural pattern recognition. Suppose that two region boundaries x and y are coded into strings denoted $x_1 x_2 .... x_n$ and $y_1 y_2 .... y_m$ respectively. Let α represent the number of matches between the two strings where a match occurs in the k^{th} position if $x_k = y_k$. The number of symbols that do not match is

$$\beta = \max(|x|, |y|) - \alpha \qquad ...(i)$$

Here, |arg| is the length (number of symbols) in the string representation of the argument. It can be shown that β = 0 if and only if x and y are identical.

A simple measure of similarity between x and y is the ratio

$$R = \frac{\alpha}{\beta} = \frac{\alpha}{\max(|x|, |y|) - \alpha} \qquad ...(ii)$$

Hence R is infinite for a perfect match and 0 when none of the corresponding symbols in x and y match. Because matching is done symbol by symbol the starting point on each boundary is important in terms reducing the amount of computation.

Any method that normalizes to, or near, the same starting point is helpful, so long as it provides a computational advantage over brute force matching, which consists of starting at arbitrary points on each string and then shifting one of the string and computing equation (ii) for each shift. The largest value of R gives the best match.

Q.14. Discuss about the affine matching.

Ans. In affine transformation, there are two main approaches proposed to match objects. The first alignment, also called hypothesis and test. It computes an affine transformation based on an hypothesized correspondence between an object and model basis and then verifies the hypothesis by transforming the model to image coordinates and determining the fraction of model and image points brought into correspondence. This is taken as a measure of quality of the transformation. Suppose an image I containing n feature points, alignment consists of the following steps –

for each model M. Let m be the number of model points
  for each triple of model points do
  for each triple of image points do
    hypothesize that they are in correspondence and
  compute the affine transformation based on this correspondence.
for each of the remaining m – 3 model points do
  apply that transformation
Find correspondences between the transformed
  model points and the image points.
Measure the quality of the transformation
  (based on the number of model points that are paired with image points.)

These steps are repeated for all possible groups of three model and image points, since it is known that they uniquely determine an affine transformation.

The second method, geometric hashing or indexing is a table lookup method. It consists of representing each model object by storing information-invariant

information about it in a hash. This table is compiled offline. At recognition time, similar invariants are extracted from the sensory data I and used to index the table to find possible instance of the model the indexing mechanism, referred to as geometric hashing for point set matching is based on the following invariant. The coordinates of a point into a reference frame consisting of three noncollinear points are affine invariant. The algorithm consists of a preprocessing phase and a recognition phase.

Algorithm of Preprocessing Phase –

for each model M. Let m be the number of model points
  for each triple of noncollinear model points do
    form a basis (reference frame)
    for each of the m – 3 remaining model points do
      determine the point coordinates in that basis.
      Use the triplet of coordinates (after a proper quantization)
      as an index to an entry in the hash table,
      where the pair (M, basis) is stored.

Algorithm of Recognition Phase –

Initialization

for each entry of the hash table do
  set a counter to 0
Choose three noncollinear points of I as a basis.
  for each of the n – 3 remaining image points do
    determine its coordinates in that basis.
    Use the triplet of coordinates (after a proper quantization)
    as an index to an entry in the hash table and increment the corresponding counter.
Find the pair (M, basis) that achieved the maximum
  value of the counter when summed over the hash table.

Q.15. Describe the dynamic programming.

Ans. **A number of approaches use dynamic programming to match shape contours. A shape boundary is described by a string of symbols representing boundary segments. For instance, if the segments result from the decomposition of the boundary into convex/concave parts, there might be just three symbols (for convex, concave, and straight) or more if different degrees of convexity or concavity are considered. The matching problem becomes then a string matching problem. The string matching algorithm is as follows –**

Let $X = a_0, \ldots, a_{n-1}$ and $Y = b_0, \ldots, b_{m-1}$ be two strings of symbols. Three types of edit operations, namely, insertion, detection, and change, are defined to transform X into Y.

(i) Insertion – Insert 'a symbol 'a' into a string, denoted as $\lambda \to a$ where λ is the null symbol

(ii) Deletion – Delete a symbol from a string, denoted as $a \to \lambda$.

(iii) Change – Change one symbol into another, denoted as $a \to b$.

A nonnegative real cost function $d(a \to b)$ is assigned to each edit operation $a \to b$. The cost of a sequence of edit operations that transforms X into Y is given by the sum of the costs of the individual operations. The edit distance E (X, Y) is defined as the minimum of such total costs. Let E(i, j) be the distance between the substrings $a_0, \ldots, a_i$ and $b_0, \ldots, b_j$. It is E(n, m) = E(X, Y). Let E(0, 0) = 0; then E(i, j), $0 < i < n$, $0 < j < m$ is given by

$$E(i, j) = \min \begin{cases} E(i-1, j) + e(a_i \to \lambda) \\ E(i, j-1) + e(\lambda \to b_j) \\ E(i-1, j-1) + e(a_i \to b_j) \end{cases}$$

The matching problem can be seen as one of finding an optimal nondecreasing path in the 2D table E(i, j) from the entry (0, 0) to the entry (n, m). If the elements of the table are computed horizontally from each row to the next one, then when computing E(i, j) the values that are needed have already been computed. Since it takes constant time to compute E(i, j), the overall time complexity is given by O(nm). The space complexity is also quadratic.

Q.16. Write short note on shape matching.

Ans. The first stage aligns the input shapes in the common space Σ, so that corresponding parts on different shapes can be easily compared. This stage can be divided into a global phase and a local phase. In the global phase, we jointly compute an affine transformation T_i for each shape S_i so that in the end all shapes are roughly aligned in Σ. This is done by following the principal two-step strategy of matching multiple shapes, where the first step performs pair-wise affine matching to construct a similarity graph $\mathcal{G}$ among the input shapes along with associated relative transformations $T_{(i,j)}$, $(i, j) \in \mathcal{G}$, and the second step jointly computes an affine transformation T_i for each shape by optimizing the consistency between the induced transformations $T_j^{-1} \circ T_i$ and the relative transformations $T_{(i,j)}$. Among existing formulations to this problem, we extend the MRF formulation, due to its ability to handle noisy relative transformations. The efficiency of this formulation relies on effectively sampling the transformation space of each shape. To address this issue, we introduce a reduced affine transformation model, which is sufficient to provide an initial starting point for the local phase, and which enables us to perform the MRF

optimization for each type of 1D transformation (e.g., the rotation in the xy-plane) in a sequential manner. In this case, we only need to sample a 1D space per-shape in each subproblem.

In the local phase, we proceed to jointly optimize a free-from deformation $\mathcal{F}_i$ for each shape S_i to improve the alignment. To avoid simultaneously optimizing the deformations of all input shapes in large shape collections, we introduce an objective function, which can be optimized in an alternating manner. In particular, at each step the deformation $\mathcal{F}_i$ of each shape can be optimized separately.

PRINCIPAL COMPONENT ANALYSIS, FEATURE EXTRACTION, NEURAL NETWORK AND MACHINE LEARNING FOR IMAGE SHAPE RECOGNITION

Q.17. What do you mean by principal component analysis ?

Ans. Suppose that the data to be reduced consist of tuples or data vectors described by n attributes or dimensions. Principal components analysis, or PCA (also called the Karhunen-Loeve, or K-L, method), searches for k n-dimensional orthogonal vectors that can best be used to represent the data, where $k \leq n$. The original data are thus projected onto a much smaller space, resulting in dimensionality reduction. Unlike attribute subset selection, which reduces the attribute set size by retaining a subset of the initial set of attributes, PCA "combines" the essence of attributes by creating an alternative, smaller set of variables. The initial data can then be projected onto this smaller set. PCA often reveals relationships that were not previously suspected and thereby allows interpretations that would not ordinarily result.

PCA is computationally inexpensive, can be applied to ordered and unordered attributes and can handle sparse data and skewed data. Multidimensional data of more than two dimensions can be handled by reducing the problem to two dimensions. Principal components may be used as inputs to multiple regression and cluster analysis. In comparison with wavelet transforms, PCA tends to be better at handling sparse data, whereas wavelet transforms are more suitable for data of high dimensionality.

Q.18. What do you mean by factor analysis ?

Ans. The PCA model generally used is a distribution-free method with no underlying statistical model. However, PCA can also be derived from a generative latent variable model –

Let,

$$x = Ay + n \quad ...(i)$$

where y is Gaussian, zero-mean and white, so that $E\{yy^T\} = I$, and n is zero-mean Gaussian white noise. It is now easy to formulate the likelihood function, because the density of x, given y, is Gaussian. Scaled eigenvectors of C_x are obtained as the rows of A in the maximum likelihood solution, in the limiting case when the noise tends to zero.

This approach is one of the methods for the classic statistical technique of factor analysis (FA). It is called principal factor analysis. Generally, the goal in factor analysis is different from PCA. Factor analysis was originally developed in social sciences and psychology. The model has the form of, with the interpretation that the elements of y are the unobservable factors. The elements a_{ij} of matrix A are called factor loadings. The elements of the additive term n are called specific factors, instead of noise. Let us make the simplifying assumption that the data has been normalized to zero mean.

In FA we assume that the elements of y (the factors) are uncorrelated and Gaussian, and their variances can be absorbed into the unknown matrix A so that we can assume

$$E\{yy^T\} = I \quad ...(ii)$$

The elements of n are uncorrelated with each other and also with the factors y_i; denote $Q = E\{nn^T\}$. It is a diagonal matrix, but the variances of the noise elements are generally not assumed to be equal or infinitely small, as in the special case of principal FA. We can write the covariance matrix of the observations from equation (i) as

$$E\{xx^T\} = C_x = AA^T + Q \quad ...(iii)$$

In practice, we have a good estimate of C_x available, given by the sample co-variance matrix. The main problem is then to solve the matrix A of factor loadings and the diagonal noise covariance matrix Q such that they will explain the observed covariances from equation (iii). There is no closed-form analytic solution for A and Q.

Assuming Q is known or can be estimated, we can attempt to solve A from $AA^T = C_x - Q_x$. The number of factors is usually constrained to be much smaller than the number of dimensions in the data, so this equation cannot be exactly solved; something similar to a least-squares solution should be used instead. Clearly, this problem does not have a unique solution – any orthogonal transform or rotation of $A \rightarrow AT$, with T an orthogonal matrix (for which $TT^T = I$), will produce exactly the same left-hand side. We need some extra constraints to make the problem more unique.

Now, looking for a factor-based interpretation of the observed variables, FA typically tries to solve the matrix A in such a way that the variables would

have high loadings on a small number of factors, and very low loadings on the remaining factors. The results are then easier to interpret. This principle has been used in such techniques as varimax, quartimax, and oblimin rotations.

There are some important differences between PCA, FA, and ICA. Principal component analysis is not based on a generative model, although it can be derived from one. It is a linear transformation that is based either on variance maximization or minimum mean-square error representation. The PCA model is invertible in the (theoretical) case of no compression, i.e., when all the principal components are retained. Once the principal components y_i have been found, the original observations can be readily expressed as their linear functions as $x = \sum_{i=1}^{n} y_i w_i$, and also the principal components are simply obtained as linear functions of the observations –

$$y_i = w_i^T x$$

The FA model is a generative latent variable model; the observations are expressed in terms of the factors, but the values of the factors cannot be directly computed from the observations. This is due to the additive term of specific factors or noise which is considered important in some application fields. Further, the rows of matrix A are generally not (proportional to) eigenvectors of C_x; several different estimation methods exist.

FA, as well as PCA, is a purely second-order statistical method – only covariances between the observed variables are used in the estimation, which is due to the assumption of Gaussianity of the factors. The factors are further assumed to be uncorrelated, which also implies independence in the case of Gaussian data. ICA is a similar generative latent variable model, but now the factors or independent components are assumed to be statistically independent and non-Gaussian – a much stronger assumption that removes the rotational redundancy of the FA model. In fact, ICA can be considered as one particular method of determining the factor rotation.

Q.19. Write the general procedure of principal component analysis (PCA).

Ans. The basic procedure of PCA is as follows –

(i) The input data are normalized, so that each attribute falls within the same range. This step helps ensure that attributes with large domains will not dominate attributes with smaller domains.

(ii) PCA computes k orthonormal vectors that provide a basis for the normalized input data. These are unit vectors that each point in a direction perpendicular to the others. These vectors are referred to as the principal components. The input data are a linear combination of the principal components.

(iii) The principal components are sorted in order of decreasing "significance" or strength. The principal components essentially serve as a new set of axes for the data, providing important information about variance. That is, the sorted axes are such that the first axis shows the most variance among the data, the second axis shows the next highest variance, and so on. For example, fig. 5.13 shows the first two principal components, Y_1 and Y_2, for the given set of data originally mapped to the axes X_1 and X_2. This information helps identify groups or patterns within the data.

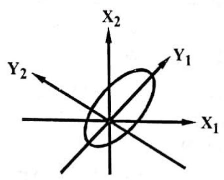


Fig. 5.7

(iv) Because the components are sorted according to decreasing order of "significance", the size of the data can be reduced by eliminating the weaker components, that is, those with low variance. Using the strongest principal components, it should be possible to reconstruct a good approximation of the original data.

Q.20. Explain PCA with example.

Ans. Eigenvalues and eigenvectors are the two most important concepts in PCA. We think the matrix as a space, then for a symmetric positive definite matrix A, its eigenvectors are orthogonal to the space formed by A. The larger the eigenvalue is, the more information corresponding eigenvector can have on this basis. For example, we select the feature values of the 10 largest eigenvalue of an image to reconstruct the image as shown in fig. 5.8, the graph is decomposed into 10 images (each feature vector corresponds to one image).

Fig. 5.8 The 10 Images Corresponding to the First 10 Principle Eigenvector

If the 10 images are superimposed, we can get a picture with not much difference to the original one, the space composed by the 10 feature vector contains 95% energy of the original image.

The idea of PCA is to reduce the data dimension to the dimensions that make the variance of the data distribution maximum. Those dimensions that make the largest variance in the data distribution are the principal components. The following steps will show how to make the maximum variance and how to select these directions –

(i) Moving the data center to 0 co-ordinates and normalize the data, that is, all the sample points X minus the mean and divided by the variance to obtain the normalized data.

(ii) Assume that the space after dimension reduction is P dimension, and now each sample point is projected to the P dimensional space, it is necessary to multiply the projection matrix P, set the P dimensional space orthogonal basis is U = [U1, U2, ..., Up]. The projection matrix is P = $(U^TU)^{-1}U^T$, U is an orthogonal matrix, so P = U^T, sample point Xi is project to the P dimensional space, the co-ordinates obtained in P dimensional space is y = Px (I) = U^Tx (I).

(iii) Now we get the co-ordinates of each sample point in the lower dimensional space, then we need to compute the variance, obviously, data variance after dimension reduction is $\sum_i y_i^T y^i$, the value y in step 2 can be carried into $\sum_i y_i^T y^i$, then the variance of the data distribution can be represented as $U^T \Sigma U$ after computation, in which Σ is the covariance matrix of X.

Now we get a lower dimensional space, the matrix composed by P orthogonal basis of the space is U, the U represent the space. As for each new sample point x, we only need to project x to U, then our goal of dimension reduction can be achieved. Therefore, the vector after dimension reduction is y = U^Tx.

Q.21. What is feature extraction ? Discuss advantages of the feature extraction.

Ans. Feature extraction is an essential process for addressing the machine learning problems. Features extraction is an essential one for the implementation of decision support system as it identifies abnormal one through selecting the essential features. The feature extraction techniques aimed on global structure for dimensionality reduction. Feature extraction is used to encode the high dimensional data into low dimensional space. The feature extraction results are enhanced by constructing set of application-dependent features called feature engineering. Feature engineering is an informal topic, but it is considered essential in applied machine learning.

Features extraction performs some transformation of original features to generate other features that are more significant "Feature extraction is generally used to mean the construction of linear combinations αTx of continuous features which have good discriminatory power between classes". An important problem in Neural Networks research and other disciplines like Artificial Intelligence is facing in finding a suitable representation of multivariate data. Features extraction can be used in this context to reduce complexity and give a simple representation of data representing each variable in feature space as a linear combination of original input variable. The most popular and widely used feature extraction approach is Principle Component Analysis (PCA) introduced by Karl. Many variants of PCA have been proposed. PCA is a simple non-parametric method used to extract the most relevant information from a set of redundant or noisy data. PCA is a linear transformation of data that minimizes the redundancy (measured through covariance) and maximizes the information (measured through the variance).

Advantages of Feature Extraction – Feature extraction is the process of extracting the relevant features from large database for dimensionality reduction. Feature extraction is a key process to reduce the dimensionality of medical dataset for efficient disease prediction. The feature extraction technique removes irrelevant features to acquire higher prediction accuracy during disease diagnosis.

In feature extraction, size of the feature space can often be decreased without losing a lot of information of the original feature space.

Q.22. Explain the principle of SVD.

Ans. Let us consider a second form of matrix analysis that leads to a low-dimensional representation of a high-dimensional matrix. This approach, called singular-value decomposition (SVD), allows an exact representation of any matrix, and also makes it easy to eliminate the less important parts of that representation to produce an approximate representation with any desired number of dimensions.

We explore the idea that the SVD defines a small number of " concepts" that connect the rows and columns of the matrix.

Let M be an m × n matrix, and let the rank of M be r. Recall that the rank of a matrix is the largest number of rows (or equivalently columns) we can choose for which no nonzero linear combination of the rows is the all-zero vector 0 (we say a set of such rows or columns is independent).

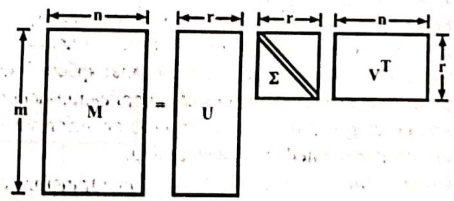


Fig. 5.9 The Form of a Singular-value Decomposition

Then, we can find matrices U, Σ, and V as shown in fig. 5.9 with the following properties –

(i) U is an $m \times r$ column-orthonormal matrix; that is, each of its columns is a unit vector and the dot product of any two columns is 0.

(ii) V is an $n \times r$ column-orthonormal matrix. Note that we always use V in its transposed form, so it is the rows of V^T that are orthonormal.

(iii) Σ is a diagonal matrix; that is, all elements not on the main diagonal are 0. The elements of Σ are called the singular values of M.

Q.23. What do you mean by singular value decomposition ?

Ans. Singular value decomposition (SVD) can be looked at from three mutually compatible points of view. On the one hand, we can see it as a method for transforming correlated variables into a set of uncorrelated ones that better expose the various relationships among the original data items. At the same time, SVD is a method for identifying and ordering the dimensions along which data points exhibit the most variation. This ties into the third way of viewing SVD, which is that once we have identified where the most variation is, it is possible to find the best approximation of the original data points using fewer dimensions. Hence, SVD can be seen as a method for data reduction.

As an illustration of these ideas, consider the 2-dimensional data points in fig. 5.10. The regression line running through them shows the best approximation of the original data with a 1-dimensional object (a line). It is the best approximation in the sense that it is the line that minimizes the distance between each original point and the line. If we drew a perpendicular line from each point to the regression line, and took the intersection of those lines as the approximation of the original datapoint, we would have a reduced representation of the original data that captures as much of the original variation as possible. Notice that there is a second regression line, perpendicular to the first, shown in fig. 5.11. This line captures as much of the variation as possible along the second dimension of the original data set. It does a poorer job of approximating the original data because it corresponds to a dimension exhibiting less variation to begin with. It is possible to use these regression lines to generate a set of uncorrelated data points that will show subgroupings in the original data not necessarily visible at first glance.

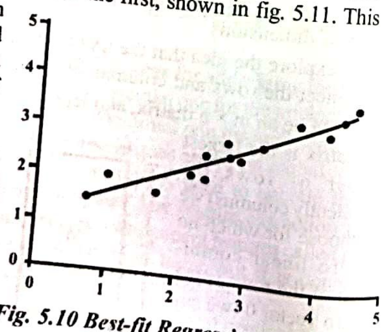


Fig. 5.10 Best-fit Regression Line Reduces Data from Two Dimensions into One

The basic ideas behind SVD that are taking a high dimensional, highly variable set of data points and reducing it to a lower dimensional space that exposes the substructure of the original data more clearly and orders it from most variation to the least. What makes SVD practical for NLP applications is that we can simply ignore variation below a particular threshold to massively reduce data but be assured that the main relationships of interest have been preserved.

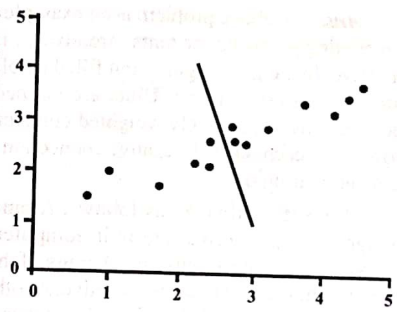


Fig. 5.11 Regression Line Along Second Dimension Captures Less Variation in Original Data

Q.24. How does a neural network work ? Explain.

Ans. A neural network can be thought of as a black box that transforms the input vector x to the output vector y, where the transformation performed is the result of the pattern of connections and weights, that is, according to the values of the weight matrix W.

Consider the vector product

$$x^*w = \Sigma x_i w_i$$

There is a geometric interpretation for this product. It is equivalent to projecting one vector onto the other vector in n-dimensional space.

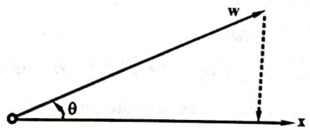


Fig. 5.12 Vector Multiplication is like Vector Projection

This notion is depicted in fig. 5.12 for the two-dimensional case.

The magnitude of the resultant vector is given by

$$x^*w = |x|\ |w|\ \cos\theta$$

where |x| denotes the norm or length of the vector x. Note that this product is maximum when both vectors point in the same direction, that is when θ = 0. The product is a minimum when both point in opposite directions or when θ = 180°.

This illustrates how the vectors in the weight matrix W influence this inputs to the nodes in a neural network.

Q.25. Find all possible stable states of the neural network.

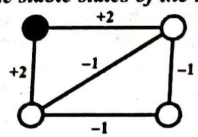


Fig. 5.13 ***(R.G.P.V., June 2013)***

Ans. The above problem is an example of a simple Hoffield net. In which processing elements, or units, are always in one of the two states, active or inactive. In the given figure, unit filled by black color is active and units which are blanked are inactive. Units are connected to each other with weighted connections. A positively weighted connection shows that the two units tend to activate each other. A negative connection permits an active unit to deactivate a neighbouring unit.

The network works as follows. A random unit is selected. If any of its neighbours are active, the unit computer the sum of the weights on the connections to those active neighbours. If the sum is positive, the unit becomes active, otherwise it becomes inactive. Another random unit is selected, and the process repeats until the network reaches a stable state, i.e., until no more units can change state.

Thus by following above procedure, we can determine all the possible stable states of the given figure.

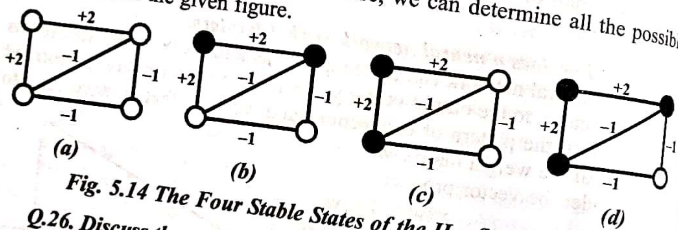


Fig. 5.14 The Four Stable States of the Hopfield Network

***Q.26. Discuss the applications of neural network.* (R.G.P.V., June 2009)**

Or

Write short note on applications of neural network.

(R.G.P.V., June 2010, 2011, May 2018)

Or

***Give any three applications of neural network.* (R.G.P.V., June 2016)**

Ans. The various applications of neural networks are discussed as below–

(i) Connectionist Speech – Speech recognition is a difficult perceptual task. Connectionist networks have been applied to a number of problems in speech recognition. Fig. 5.15 shows how a three-layer back propagation network can be trained to discriminate between different vowel sounds. The network is trained to output one of ten vowels, given a pair of frequencies taken from the speech waveform.

Speech Production – The problem of translating text into speech rather than viceversa – has also been attacked with neural networks. Speech production is easier than speech recognition, and high performance programs are available. NET talk, a network that learns to pronounce English text, was one of the first systems to demonstrate that connectionist methods could be applied to real-world tasks.

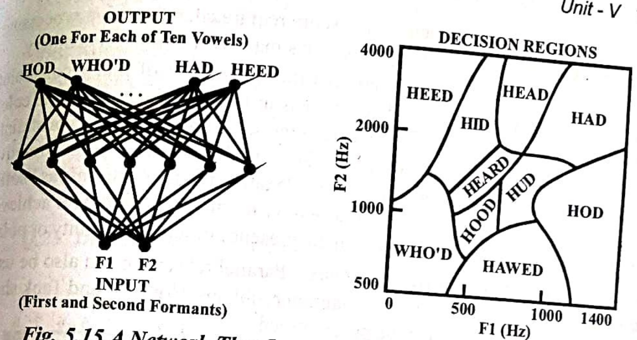


Fig. 5.15 A Network That Learns to Distinguish Vowel Sounds

Linguists have long studied the rules governing the translation of text into speech units called *phonemes*. For example, the letter "x" is usually pronounced with as "ks" sound, as in "box" and "are". A traditional approach to the problem world be to write all these rules down and use a production system to apply them. Unfortunately, most of the rules have exceptions – consider "xylophone" – and these exceptions must also be programmed in. Also, the rules may interact with one another in unpleasant, unforeseen ways. A connectionist approach is simply to present a network with words and their pronunciations, and hope that the network will discover the regularities and remember the exceptions. NET talk succeeds fairly well at this task with a back propagation network.

(ii) Connectionist Vision – Humans achieve significant visual prowess with limited hardware. Only the center of the retina maintains good spatial resolution; as a result, we must constantly shift our attention among various points of interest. Each snapshot lasts only about two hundred milliseconds. Since individual neural firing rates usually lie in the millisecond range, each scene must be interrupted in about a hundred computational steps. To compound the problem, each interpretation must be rapidly integrated with previous interpretations to enable the construction of a stable three-dimensional model of the world. These severe timing constraints strongly suggest that human vision is highly parallel. Connectionism offers many methods for studying both the engineering and biological aspects of massively parallel vision.

Parallel relaxation plays an important role in connectionist vision systems. In a typical system, some neural units receive their initial activation levels from a video camera and then these activations are iteratively modified based on the influences of nearly units one use for relaxation is detecting edges. If many units think they are located on an edge border, they can override any dissenters. The relaxation process settles on the most likely set of edges in the scene. While traditional vision programs running on sexual computing engines

must reason about which regions of a score require edge detection processing the connectionist approach simply assumes massively parallel machinery.

Visual interpretation also requires the integration of many constraint sources. For example, if two adjacent areas in the score have the same color and texture, then they are probably part of the same object. If these constraints can be encoded in a network structure, then parallel relaxation is an attractive technique for combining them. Because relaxation treats constraints as "soft", i.e., it will violate one constraint if necessary to satisfy the others. It achieves a global best-fit interpretation even in the presence of local ambiguity or noise.

(iii) Combinatorial Problems – Parallel relaxation can also be used to solve many other constraint satisfaction problems. Hopfield and Tank show how a Hopfield network can be programmed to come up with approximate solutions to the traveling salesman problem. The system employs n^2 neural units, where n is the number of cities to be toured. Fig. 5.16 shows how tours themselves are represented. Each row stands for one city. The tour proceeds horizontally across the columns. The starting city is marked by the active unit in column 1, the next city by column 2, etc. The tour shown in fig. 5.16 goes through cities D, B, E, H, G, F, C, A, and back to D.

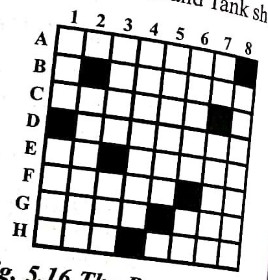


Fig. 5.16 The Representation of a Travelling Salesman Tour in a Hopfield Network

Like all Hopfield networks, this n by n array contains a number of weighted connections. The connection weights are initialized to reflect exactly the constraints of a particular problem instance. First of all, every unit is connected with a negative weight to every other unit in this column, because only one city at a time can be visited. Second, every unit inhibits every other unit in its row, because each city can only be visited once. Third, units in adjacent columns inhibit each other in proportion to the distances between cities represented by their rows. For example, if city D is far from city G, then the fourth unit in column 3 will strongly inhibit the seventh units in columns 2 and 4. There is some global excitation, so in the absence of strong inhibition, individual units will prefer to be active.

Notice that each unit represents some hypothesis about the position of a particular city in a short tour. To find that tour, we start out by giving our units random activation values. Once all the weights are set, the units update themselves asynchronously according to the rule. This updating continues until a stable state is reached. Stable states of the network correspond to short tours because conflicts between constraints are minimal.

Other tasks successfully tackled by neural networks, include learning to play backgrammar to classify sonar signals, to compress images, and to drive a vehicle along a road.

Q.27. Briefly explain the applications of neural networks. What are the drawbacks of neural network ? ***(R.G.P.V. Dec. 2009)***

Ans. **Applications** – Refer Q.26.

Drawbacks – Drawbacks of neural network are as follows –

(i) The neural network needs training to operate.

(ii) The architecture of a neural network is different from the architecture of the microprocessor therefore needs to be emulated.

(iii) Requires high processing time for large neural networks.

Q.28. Discuss the applications of neural network in medical field. ***(R.G.P.V., Nov. 2018)***

Ans. The applications of neural network in medical field are as follows –

(i) Cardiology – Serum enzyme level analysis forms the basis of acute myocardial infarction (AMI) diagnostics. A neural network has been trained for the analysis of these heart enzyme levels. Diagnostic accuracy proved to be 100% with an 8% false positive rate. Later, the same research group developed an integrated decision support system in which a neural network was trained not only by enzymatic data, but also by EKG-phenomena, subjective symptoms and changes after administration of nitroglycerine.

Neural networks were used to study the sophisticated control of cardioverter defibrillators. Neural networks have been used to model heart rate regulation but ortizetal used them to examine heart failure.

(ii) Oncology – There are several systems available for the diagnosis and selection of therapeutic strategies in breast cancer. A neural network judged the possible recurrence rate of tumors correctly in 960 of 1008 cases by using data from lymphatic node positive patients (tumor size, tumor hormone receptor status, number of palpable lymphatic nodules, etc.). Neural network recognition of breast cancer that evaluation of mammographic, cytological and epidemiological findings in an integrated system is thought to be useful in the diagnostic process.

(iii) Neurology – The sometimes difficult differential diagnosis between Alzheimer disease and vascular dementia can be assisted by neural network analysis of brain SPECT image data.

(iv) Pulmonology – Pulmonologists and radiologists have worked together on the development of a system for the classification of solitary pulmonary nodules. According to their results, neural network analysis of such disorders was less successful than conventional classification methods. In contrast, neural networks were more accurate than 2 well trained experts for the diagnosis of pulmonary embolism.

(v) Radiology – To date, the application of neural networks seems to be most interesting and most powerful in the field of radiology. Images contain

much information and they are so complicated that it's all but impossible to interpret then using conventional rule based systems. By selecting an appropriate training set and learning process, neural network modeling becomes suitable for noise filtering and for recognition of unusual images.

Abdominal ultrasound and laboratory investigations do not usually provide enough data for the differentiation of liver diseases. Based on ultrasonographic and laboratory findings, a neural network was created to diagnose five classes of liver diseases. The network achieved a recognition accuracy somewhere between the results of residents and those of certified radiologists.

(vi) Otorhinolaryngology – Neural networks have proven to be a new and effective method for modeling hearing. This technique could become useful for understanding, modeling and treating speech and hearing impairments. Hearing-aids can well be improved by using neural networks for noise filtering and optimization of parameter settings.

Table 5.2 shows some other applications of neural networks in medical fields.

Table 5.2 Applications of Neural Networks

Discipline	*Application Field*
Cardiology	Diagnostics, Prognostics
ECG	Diagnostics
Intensive care	Prediction
Gastroenterology	Prediction
Pulmonology	Diagnostics
Oncology	Diagnostics, Prognostics
Paediatrics	Diagnostics
Neurology	Signal processing, Modelling
EEG	Diagnostics
Otology-Rhinology-Laryngology	Signal processing, Modelling
Obstetrics and Gynaecology	Prediction
Ophthalmology	Signal processing, Modelling
Radiology	Signal processing (X-ray, US, CT)
Clinical chemistry	Signal processing, Diagnostics
Pathology	Diagnostics, Prognostics
Cytology	Diagnostics, Re-screening
Genetics	Diagnostics
Biochemistry	Protein sequence, Structure

Q.29. Discuss about the convolutional neural networks (CNNs).

Ans. The most popular neural network in the field of computer vision is a convolutional network, it is in use for a long time and got popularity in recent years as the development of hardware have empowered machines with more computational power the convolutional networks are going towards deep learning providing better results. For the object detection purpose the most popular and used deep learning models consist of the recurrent convolutional network.

Convolutional Neural Networks (CNNs) are much successful in various machine learning and computer vision problems. These are used in variety of areas in recognition and robotic field. There are a number of reasons that convolutional neural networks (CNNs) are becoming important. In the meantime convolutional neural networks (CNNs) have been applied with great success to the image recognition of objects in the air from the past few years. From the sides of validity it is more naturally and simply in term of capturing an image.

CNNs assumes the nature of the image, such as static images and where pixel dependencies are. Thus, compared to standard feed-forward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters and that is why they are easier to train. In addition, the capacity of CNNs can be controlled by varying their depth and breadth due to their architecture performance.

Consequently, the typical architecture of CNNs is a multilayer stack of simple modules such as convolutional layer, pooling layer and fully-connected layer. Starting with the raw input, each module transforms the representation at one level into a higher and more abstract level. Meanwhile, for recognition tasks, higher ranking of dataset representations an increase aspects of the input which are important for discrimination and limit unrelated of variations.

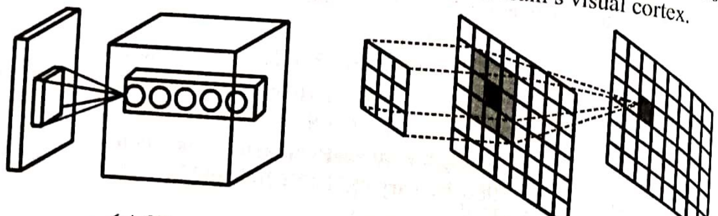
The deep learning technique based on convolutional neural network has achieved great performance improvement in large-scale image classification tasks and set off the upsurge of deep learning besides it is a new hot spot in the domain of pattern recognition. It allows a model that consists of multiple processing layers to study data representation with various levels of abstraction. Furthermore, in deep learning techniques, besides data formation, transfer learning is useful when someone wants to train on their own dataset for various reasons, for example, the dataset may not be enough to train the full neural net and cause problems in transfer learning.

Specifically, transfer learning may be used to take a pre-trained deep neural networks, replacing the fully-connected layers (and potentially the last convolutional layer) and training those layers on the related dataset. Nevertheless, it has been observed that deep neural networks (DNNs) easily suffer from over fitting with small samples. In this review, the technique of machine learning

is important to ensure the quality and efficiency of image in terms of capturing, verification and clustering that want to train more effectiveness especially in image recognition by using the accurate learning machines method.

Q.30. *Explain in detail about the convolution layer.*

Ans. The convolutional layer is the core building block of a convolutional neural network and aims to resolve the limitations of fully connected layers by making geometric assumptions about the input data. The layer's parameters consist of a set of learnable filters or kernels, which have a small receptive field, but extend through the full depth of the input volume. The architecture found in convolutional layers – or convolutional networks in general – is loosely based on the complex arrangement of cells in the mammalian brain's visual cortex.



(a) 3D *(b) 2D*

Fig. 5.17 Input, Filter and Output of a Convolution in 2D and 3D

During the forward pass, the learnable filters are convolved with the input volume. The intuition behind the discrete convolution operation is to slide a filter over the input volume at different overlapping spatial locations as in

$$\mathbf{x}_f^{(\ell)} = \sum_{u,v} \mathbf{x}_{uv}^{(\ell-1)} * \mathbf{w}_f^{(\ell)} + \mathbf{b}_f^{(\ell)}$$

where $\mathbf{x}_f^{(\ell)}$ represents the current layer's output for a given filter f, $\mathbf{x}^{(\ell-1)}$ the output of the previous layer, and the spatial extent of the filter in horizontal and vertical direction is given by u and v.

During the backward pass, we calculate the partial derivatives of the loss function with respect to the weights and biases of the respective layer as in

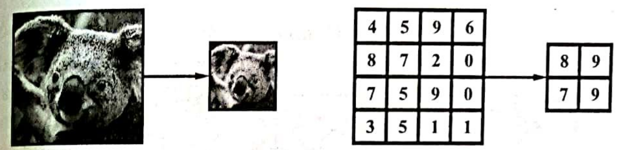
$$\nabla_{\mathbf{w}_f^{(\ell)}} \mathcal{L} = \sum_{u,v} \left(\nabla_{\mathbf{x}_f^{(\ell+1)}} \mathcal{L} \right)_{uv} \left(\mathbf{x}_{uv}^{(\ell)} * \hat{\mathbf{w}}_f^{(\ell)} \right)$$

$$\nabla_{\mathbf{b}_f^{(\ell)}} \mathcal{L} = \sum_{u,v} \left(\nabla_{\mathbf{x}_f^{(\ell+1)}} \mathcal{L} \right)_{uv}$$

where, $\hat{\mathbf{w}}$ denotes a spatially flipped filter in order to compute the cross-correlation rather than a convolution.

Q.31. *What do you mean by pooling layer ? Explain.*

Ans. Another important concept of convolutional networks is pooling, which is a form of non-linear downsampling. The pooling layer partitions the input volume into a set of non-overlapping rectangles and, for each subregion, outputs the maximum activation, hence the name max-pooling as shown in fig. 5.18 Another common pooling operation is average pooling, which computes the mean of the activations in the previous layer rather than the maximum. The function of the pooling layer is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. It is common practice to periodically insert a pooling layer in between successive convolutional layers.



(a) Global *(b) Local*

Fig. 5.18 Max-pooling Layer

During the forward pass, the maximum of non-overlapping regions of the previous activations is computed as in

$$\mathbf{x}^{(\ell)} = \max_{u,v}\left(\mathbf{x}^{(\ell-1)}\right)_{uv}$$

where u and v denote the spatial extent of the non-overlapping regions in width and height.

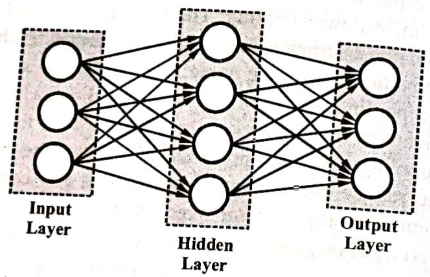
Since the pooling layer does not have any learnable parameters, the backward pass is merely an upsampling operation of the upstream derivatives. In case of the max-pooling operation, it is common practice to keep track of the index of the maximum activation so that the gradient can be routed towards its origin during back propagation.

The hyperparameters of the pooling layer are its stride and filter size. Since they have to be chosen in accordance to each other, they can be interpreted as the amount of downsampling to be performed.

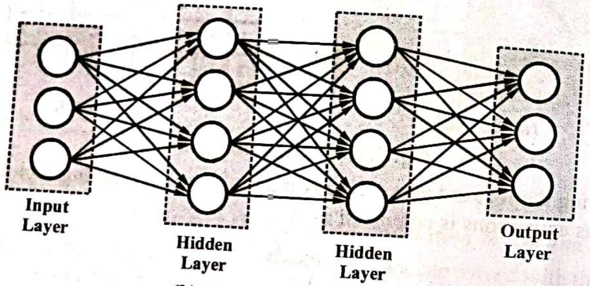
Q.32. *What is fully connected layer ? Discuss its limitation.*

Ans. The fully connected layer is a synonym often used in the convolutional network literature and is equivalent to a hidden layer in a regular artificial network. It is sometimes referred to as linear or affine layer. Intuitively, the fully connected layer is responsible for the high-level reasoning in a convolutional neural network and is therefore typically inserted after the convolutional layers. The neurons have full connections to all activations in

the previous layer, as seen in a regular artificial neural network as shown in fig. 5.19.



(a) One Hidden Layer



(b) Two Hidden Layers

Fig. 5.19 Fully Connected Layers in an Artificial Neural Network

The activations of a fully connected layer can be computed with a dot product of the weights with the previous layer activations followed by a bias offset as in

$$\mathbf{x}^{(\ell)} = (\mathbf{w}^{(\ell)})^T \mathbf{x}^{(\ell-1)} + \mathbf{b}^{(\ell)}$$

where $\mathbf{x}^{(\ell-1)}$ denotes the activations of the previous layer and $\mathbf{x}^{(\ell)}$, $\mathbf{w}^{(\ell)}$, and $\mathbf{b}^{(\ell)}$ denote the activations, weights and biases of the current layer, respectively.

During the backward pass, the gradient with respect to the weights and biases are computed as in

$$\nabla_{\mathbf{w}^{(\ell)}} \mathcal{L} = (\mathbf{x}^{(\ell)})^T (\nabla_{\mathbf{x}^{(\ell+1)}} \mathcal{L})$$

$$\nabla_{\mathbf{b}^{(\ell)}} \mathcal{L} = \sum_{i=1}^{n} (\nabla_{\mathbf{x}^{(\ell+1)}} \mathcal{L})_i^T$$

where $\nabla_{\mathbf{x}^{(\ell+1)}}$ denotes the upstream derivatives.

The only hyperparameter in a fully connected layer is the number of output neurons the input connects to, i.e. how many learnable parameters connect the input to the output.

The main limitation of the fully connected layer is the assumption that each input feature, i.e. pixel in the image, is completely independent of neighbouring pixels and contributes equally to the predictive performance. However, pixels that are close together have the tendency to be highly correlated and thus the spatial structure of images has to be taken into account. Additionally, fully connected layers do not scale well to high dimensional data such as images since each pixel of the input has to be connected to the layer's output with a learnable parameter.

Q.33. What is machine learning ?

Ans. Machine learning is a branch of science that deals with programming the systems in such a way that they automatically learn and improve with experience. Here, learning means recognizing and understanding the input data and making wise decisions based on the supplied data.

It is very difficult to cater to all the decisions based on all possible inputs. To tackle this problem, algorithms are developed. These algorithms build knowledge from specific data and past experience with the principles of statistics, probability theory, logic, combinatorial optimization, search, reinforcement learning and control theory.

The developed algorithms from the basis of various applications such as–

(i) Vision processing
(ii) Language processing
(iii) Forecasting (e.g., stock market trends)
(iv) Pattern recognition
(v) Games
(vi) Data mining
(vii) Expert systems
(viii) Robotics.

Q.34. Write and explain different types of machine learning.

Ans. The different types of machine learning are shown in fig. 5.20.

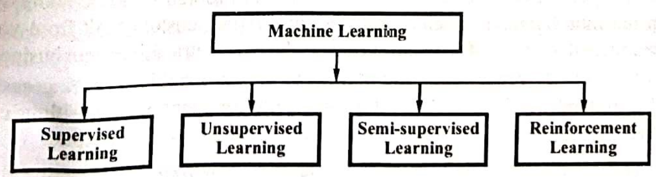


Fig. 5.20 Types of Machine Learning

(i) ***Supervised Learning*** – In this type of learning, the machine is provided with a given set of inputs with their desired outputs. The machine

needs to study those given sets of inputs and outputs and find a general function that maps inputs to desired outputs.

(ii) Unsupervised Learning – This type of learning is termed as 'learned by its own' by discovering and adopting, based on the input pattern. In this learning the data are divided into different clusters and hence the learning is called a ***clustering algorithm.***

(iii) Semi-supervised Learning – This learning is used for the same applications as supervised learning. But it uses both labeled and unlabeled data for training. This type of learning can be used with methods such as classification, regression and prediction. Semi-supervised learning is useful when the cost associated with labeling is too high to allow for a fully labeled training process. Early examples of this include identifying a person's face on a web cam.

(iv) Reinforcement Learning (RL) – In this type of learning, machine is trained to take specific decisions based on the business requirement with the objective to maximize the efficiency (performance). This continual learning process ensures less participation of human expertise and saves more time. Reinforcement learning is often used for robotics, gaming and navigation. With reinforcement learning, the algorithm discovers through trial and error which actions yield the greatest rewards.

Q.35. Write the advantages of machine learning.

Ans. The five advantages of machine learning are as follows –

(i) Accurate – Machine learning uses data to discover the optimal decision making engine for your problem. As you collect more data, the accuracy can increase automatically.

(ii) Automated – As answers are validated or discarded, the machine learning model can learn new patterns automatically. This allows users to embed machine learning directly into an automated workflow.

(iii) Fast – Machine learning can generate answers in a matter of milliseconds as new data streams in, allowing systems to react in real time.

(iv) Customizable – Many data-driven problems can be addressed with machine learning. Machine learning models are custom built from your own data, and can be configured to optimize whatever metric drives your business.

(v) Scalable – As your business grows, machine learning easily scales to handle increased data rates. Some machine learning algorithms can scale to handle large amounts of data on many machines in the cloud.

Q.36. Write the disadvantages of machine learning.

Ans. The disadvantages of machine learning are as follows –

(i) Machine learning has the major challenge called acquisition. Also based on different algorithms data need to be processed. And, it must be

processed before providing as input to respective algorithms. Thus, it has a significant impact on results to be achieved or obtained.

(ii) As we have one more term interpretation. That it result is also a major challenge. That need to determine the effectiveness of machine learning algorithms.

(iii) We can say uses of machine algorithm is limited. Also, it's not having any surety that it's algorithms will always work in every case imaginable. As we have seen that in most cases machine learning fails. Thus, it requires some understanding of the problem at hand to apply the right algorithm.

(iv) Like deep learning algorithm, machine learning also needs a lot of training data. As we can say it might be cumbersome to work with a large amount of data. Fortunately, there are a lot of training data for image recognition purposes.

(v) One notable limitation of machine learning is its susceptibility to errors. Brynjolfsson and McAfee said that the actual problem with this inevitable fact. That when they do make errors, diagnosing and correcting them can be difficult. As because it will need going through the underlying complexities.

Q.37. Explain the applications of machine learning.

Ans. (i) Computer Vision – Many current vision systems, from face recognition systems, to systems that automatically classify microscope images of cells, are developed using machine learning, again because the resulting systems are more accurate than hand-crafted programs. One massive-scale application of computer vision trained using machine learning is its use by the US Post Office to automatically sort letters containing handwritten addresses. Over 85% of handwritten mail in the US is sorted automatically, using handwriting analysis software trained to very high accuracy using machine learning over a very large data set.

(ii) Speech Recognition – Currently available commercial systems for speech recognition all use machine learning in one fashion or another to train the system to recognize speech. The reason is simple – the speech recognition accuracy is greater if one trains the system, then if one attempts to program it by hand. In fact, many commercial speech recognition systems involve two distinct learning phases – one before the software is shipped (training the general system in a speaker-independent fashion), and a second phase after the user purchases the software (to achieve greater accuracy by training in a speaker-dependent fashion).

(iii) Bio-surveillance – A variety of government efforts to detect and track disease outbreaks now use machine learning. For example, the RODS project involves real-time collection of admissions reports to emergency rooms across western Pennsylvania, and the use of machine learning software to learn the profile of typical admissions so that it can detect anomalous patterns of symptoms and their geographical distribution. Current work involves adding in

a rich set of additional data, such as retail purchases of over-the-counter medicines to increase the information flow into the system, further increasing the need for automated learning methods given this even more complex data set.

(iv) Robot Control – Machine learning methods have been successfully used in a number of robot systems. For example, several researchers have demonstrated the use of machine learning to acquire control strategies for stable helicopter flight and helicopter aerobatics. The recent Darpa-sponsored competition involving a robot driving autonomously for over 100 miles in the desert was won by a robot that used machine learning to refine its ability to detect distant objects (training itself from self-collected data consisting of terrain seen initially in the distance, and seen later up close.

(v) Natural Language Processing – It is a field that which involves both computer understanding and manipulation of human language and its good in gathering new possibilities. It is mostly seen in a large pool of legislation or other document sets, trying to discover new patterns or to root out corruption. It is a better way to analyze, understand and find the meaning of human language easily and smartly. By using NLP developer can perform tasks such as speech recognition, entity recognition, automatic translation, and summarization.

Discrete event simulation is a technique where patients are modeled as an independent even associating each with some attribute information like age, weight and problematic scenarios, etc. Natural language processing is a technique used for system to read the physician's notes and convert it to digital data. Proprietary predictive model is used to make predictions such as admissions can be predicted by hospitals to spread expertise which is in short supply. Disease prediction and diagnosis is achieved by helping radiologists to make intellectual decisions with radiology data (for example – CT, MRI and Radiographs).

Q.38. What are machine learning tools ? Explain.

Ans. Machine learning gives a set of tools that use computers to transform data into actionable information. Tools are a big part of machine learning and choosing the right tool can be as important as working with the best algorithms. Machine learning tools make applied machine learning faster, easier. Good tools can automate each step in the applied machine learning process by shortening the time.

The machine learning tools are as follows –

(i) Platforms – Platforms are used to complete machine learning project from beginning to end.

(a) Provide capabilities required at each step in a machine learning project.

(b) The interface may be graphical or command line.

(c) They provide a lose coupling of features.

(d) They are provided for general purpose use and exploration rather than speed, scalability or accuracy.

(ii) Library – Library gives capabilities for completing part of a machine learning project.

(a) Provide a specific capability for one or more steps in a machine learning project.

(b) The interface is typically an application programming interface requiring programming.

(c) They are tailored for a specific use case, problem type or environment.

(iii) Graphical User Interfaces –

(a) Allows less-technical users to work through machine learning.

(b) Focus on process and how to get the most from machine learning techniques.

(c) Stronger focus on graphical presentations of information such as visualization.

(d) Structured process imposed on the user by the interface.

(iv) Command Line Interface –

(a) Allows technical users who are not programmers to work through machine learning projects.

(b) Frames machine learning tasks in terms of the input required and output to be generated.

(c) Promotes reproducible results by recording or scripting commands and command line arguments.

(v) Application Programming Interfaces –

(a) To incorporate machine learning into our own software projects.

(b) To create our own machine learning tools.

(c) Gives the flexibility to use our own processes and automations on machine learning projects.

(d) Allows combining our own methods with those provided by the library as well as extending provided methods.

(vi) Local Tools – Local tools can be downloaded, installed and run on local environment.

(a) Customized for in-memory data and algorithms.

(b) Control over run configuration and parameterization.

(c) Integrate into our own systems to meet our needs.

(vii) Remote Tools – Remote tools can be hosted on a server and called from local environment. These tools are often referred to as machine learning as a service (MLaaS).

(a) Tailored for scale to be run on larger datasets.

(b) Run across multiple systems, multiple cores and shared memory.

Q.39. Write and explain scope of machine learning.

Ans. The scope of machine learning are as follows –

(i) Explaining Human Learning – A mentioned earlier, machine learning theories have been preceived fitting to comprehend features of learning in humans and animals. Reinforcement learning algorithms estimate the dopaminergic neurones induced activites in animals during reward-based learning with surprising accuracy. ML algorithms for uncovering sporadicdelineations of naturally appearing images predict visual features detected in animals initial visual cortex. Nevertheless, the important drivers in human or animal learning like stimulation, horror, urgency, hunger, instinctive actions and learning by trial and error over numerous time scales, are not yet taken into account in ML algorithms. This a potential opportunity to discover a more generalised concept of learning that entailsboth animals and machine.

(ii) Programming Languages Containing Machine Learning Primitives – In majority of applications, ML algorithms are incorporated with manually coded programs as part of an application software. The need of a new programming language that is self-sufficient to support manually written subroutines as well as those defined as "to be learned". Programming languages like Python (Sckit-learn), R etc. already making use of this concept in smaller scope. But a fascinating new question is raised as to develop a model to define relevant learning experience for each subroutines tagged as "to be learned", timing, and security in case of any unforeseen modification to the program's function.

(iii) Perception – A generalised concept of computer perception that can link ML algorithms which are used in numerous form of computer perception today including but not limited to highly advanced vision, speech recognition etc., is another potential research area. One thought-provoking problem is the integration of different senses (e.g., sight, hear, touch etc.) to prepare a system which employ self-supervised learning to estimate one sensory knowledge using the others. Researches in developmental psychology have noted more effective learning in humans when various input modalities are supplied, and studies on co-training methods in sinuate similar results.

☯ ☯ ☯